INTERPRETATION, GROUNDING AND IMAGINATION FOR MACHINE INTELLIGENCE

A Dissertation
Presented to
The Academic Faculty

By

Shanmukha Ramakrishna Vedantam

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Interactive Computing

Georgia Institute of Technology

December 2018

INTERPRETATION, GROUNDING AND IMAGINATION FOR MACHINE INTELLIGENCE

Approved by:

Dr. Devi Parikh
School of Interactive Computing
Georgia Institute of Technology

Dr. Dhruv Batra
School of Interactive Computing
Georgia Institute of Technology

Dr. Jacob Eisenstein
School of Interactive Computing
Georgia Institute of Technology

Dr. C. Lawrence Zitnick
Facebook AI Research

Dr. Kevin P. Murphy
Google Research

Date Approved: October 24, 2018

All our knowledge begins with the senses,

proceeds then to the understanding, and ends with reason.

There is nothing higher than reason.

Immanuel Kant

To Amma, Baba, Suvarchala pinni and my late grandfather, Gullapalli Krishna Das.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my parents, Vedantam Sasi Bhushana Sarma and Vedantam Kanaka Durga, for all the love, encouragement and support and being pillars of strength through all times, good and the bad. Thank you for understanding when I have been unresponsive or distracted, and for sending me all the way across the other side of the world in search of a flimsy, utopian dream. This thesis is as much your work as it is mine.

Likewise, it is hard to find words to express how grateful I am to my advisor, Devi Parikh. During the last five years Devi is the one presence in my life that I have found consistently convenient to look up to, as a sounding board for when things go wrong, and to be the voice of reason and reasonability in any conversation.

Thank you for teaching me first hand the values of doing curiosity driven, original and creative research, and for giving me the freedom and flexibility to find out what I am curious about. As I reflect on the person I was five years ago, and think to this date, it is hard to find an aspect of my life that has not been changed, in some profound way through our interactions.

I would also like to thank Dhruv Batra, for all the advice and support through the years, and for always being the devil's advocate on my first few papers in grad school. I knew if I could convince Dhruv, I could convince any of the three anonymous reviewers! On a serious note, I would like to thank Dhruv for teaching amazingly accessible and fun classes on machine learning, and probabilistic graphical models at Virginia Tech. Thank you for introducing to a naive undergrad out of bachelors to the worlds and joys of maximizing expectations, message passing and long-short term memories!

I would like to thank Kevin Murphy for being a wonderful mentor figure and internship advisor, shaping so many of my tastes and outlook around research. Thank you for sharing your infectious enthusiasm for probabilistic machine learning, and for being a constant reference point for the kind of researcher I dream of being some day.

I would also like to thank Ian Fischer for being a sounding board when making decisions, for teaching me how to write maintainable, high-quality research code, and for embodying the perfect example of doing principled, hypothesis driven research. I also thank Gal Chechik, and Samy Bengio for hosting me for an amazing internship at Google (where we spent as much time coding, as we spent looking at images of birds!), and Rahul Sukthankar for his generosity with giving valuable advice.

I would also like to thank Larry Zitnick for awing a young grad student with his curiosity, infectious enthusiasm and relentless creativity. Working with Larry has been one of the truly inspiring experiences of my PhD.

I would like to thank Jacob Eisenstein for agreeing to serve on my committee and for his comments on making the motivations for this work more grounded in lexical and formal semantics.

Finally, I would like to thank my labmates over the years at the Computer Vision and Machine Learning and Perception (CVMLP) Labs, first at Virginia Tech and then at Georgia Tech. Specifically, I would like to thank Shrenik Lad being an awesome flatmate, and for making the first two years of graduate school so much fun, and for being a constant presence through good and bad times. Further, I would like to thank Xiao Lin, Stanislaw Antol, Qing Sun, Qi Lou, Faruk Ahmed, Adarsh Prasad, Ankit Laddha, Peng Zhang, Senthil Purushwalkam, Ramprasaath Selvaraju, Prakriti Banik, Yash Goyal, Aishwarya Agarwal, Michael Cogswell, Akrit Mohapatra, Neelima Chavali, Clint Solomon, Mainak Jas, Harsh Agarwal, Abhishek Das, Arjun Chandrasekharan, Ashwin Kalyan, Prithvijit Chattopadhyay, Satwik Kottur, Deshraj Yadav, Viraj Prabhu, Jiasen Lu, Jianwei Yang, Stefan Lee, Samyak Datta and Nirbhay Modhe for all the brainstorming sessions over the years and for being a part of an amazing community in the CVMLP labs.

Specifically, I would also like to thank the Terrace View crew – Harsh Agarwal, Arjun Chandrasekharan, Abhishek Das, Ramprasaath Selvaraju, for being amazing roommates and collaborators! Those two years were definitely the most fun part of grad school for me.

Finally I would like to thank Prateek Shekhar, Nilaksh Das, Sanya Chaba, and Dileep Basam for being extremely supportive at different points through this journey, and Ravi Agarwal being an amazing friend, and for persistently (temporarily) pulling me out of grad school for doses of normalcy in the various visits to DC, and trips elsewhere.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Thesis Statement

Modeling the interplay between language and vision with semantic and pragmatic considerations can help derive more human-like inferences from machine learning models; specifically in making them capable of

1. Interpreting an image and describing its contents using natural language in a contextually relevant manner
2. Grounding natural language in the physical world to learn common sense
3. Imagining visual concepts completely and accurately across the full range and (potentially unseen) compositions of their visual attributes

SUMMARY

The goal of this thesis is to build machine learning approaches to derive more human-like inferences from machine learning models. We will consider various situations where humans are able to make intuitive inferences, but where machines are might not, and show how appropriate considerations about semantics or pragmatics can improve the inferences made by machines and make them more human-like.

In pursuit of this overarching goal, I will look at three focus areas: interpretation, grounding, and visual imagination.

In interpretation, I will study how to go from computer vision to natural language. Specifically, the focus will be on image captioning: the problem of describing an image with a natural language description. Here, I study how to formalize the task of language generation given images and create evaluation schemes to make progress towards generating more 'human-like' descriptions. As humans, we make pragmatic considerations frequently in our usage of language, by modeling the context of a particular situation or interaction. I will present such an incarnation of the image captioning problem, and an algorithmic solution to generate captions which are more aware of the context in which we want to describe an image.

In the second part, I will consider the inverse problem of "grounding" – learning the notion of a word such as a "car" in our lexicon in terms of what it refers to in the physical world. Capturing grounded semantics of symbols in terms of the physical world can enable machines which demonstrate an ability to make more intuitive inferences about our world. A key focus will be on learning word representations grounded in vision or sound, and a study of how such representations can lead to improved retrieval and commonsense reasoning, where the goal will be to predict if an assertion specified in text is plausible, *i.e.* happens in the real world or not.

Finally, I will focus on the problem of imagination – *i.e.*, performing conditional genera-

tion of pixels in an image given a concept that has never been seen before. As humans, we can imagine what a purple hippo would look like, even though they do not exist. Concretely, if we instead said "purple hippo with wings", we could just as easily create a different internal mental representation, to represent this more specific concept. To assess whether the person has correctly understood the implied concept, we can ask them to draw a few sketches, to illustrate their thoughts. I will call the ability to map text descriptions of concepts to latent representations and then to images visually grounded semantic imagination. In this thesis, we will assume that the words we specify compose as intersections, *i.e.*, the set of all purple hippos is roughly the intersection of all instances which are purple and all instances which are hippos. The general problem specification, of handling all possible types of combinations of words is difficult in general, and remains an open challenge.

In this chapter, we will focus on approaches that: 1) generate pixels from compositional attribute vectors (as opposed to grounding them into some abstract feature space), and 2) impose intuitive constraints on the attribute vectors to ensure that the output images match the intension (roughly, the definition of the concept), and the extension (the variance or span of the concept). Further, I will describe how a class of joint multimodal variational autoencoders can be adapted to perform this task, and other tasks that such a model family might be able to accomplish, such as providing a name for a set of images.

CHAPTER 1
INTRODUCTION

Vision and language are intricately related, and play a crucial role in shaping human intelligence. As humans, we have a striking ability to recognize and process information from high dimensional perceptual signals such as vision (Kitcher 1988), condense the information from perception into groups of concepts and categories (Rosch 1999), and more generally, express and communicate about concepts in the form of natural language. How can we build machines that are able to derive insights from vision and language and come up with inferences similar to humans?

Modeling vision and language jointly in such a manner is a grand challenge in artificial intelligence (AI). Recent years have seen exciting progress on individual aspects of this problem. On the vision side alone, we have seen impressive gains in performance across numerous tasks, starting with the problem of image classification, which is the task of assigning a category from a lexicon to an input image. This progress has been driven by a advances in training deep convolutional neural network architectures, which nowadays come in various flavors (Krizhevsky, Sutskever, and Hinton 2012; Simonyan and Zisserman 2015; Szegedy et al. 2015; He et al. 2016). These networks, when trained with stochastic gradient descent using the backpropagation (Rumelhart, Hinton, and Williams 1986) algorithm on modern commodity graphics processing units (GPUs) (which speed up training), on large scale labelled datasets such as the ImageNet dataset (Deng et al. 2009) have led to unprecedented advances in the state-of-the-art. An attractive by product of this progress has been in the realization that the features learnt from such classification networks are not just useful for image classification, but are useful as generic image features (Donahue et al. 2014).

In parallel, we have also made progress on problems such as neural machine trans-

lation (Sutskever, Vinyals, and Le 2014) using *recurrent neural network* architectures (Chapter. 2), which are powerful function approximators to model sequential data such as natural language. Armed with rich and generic image representations and better language models, we have finally been able to make progress on the long standing goal of describing natural images with natural language descriptions which appear increasingly realistic and human-like (Karpathy and Fei-Fei 2015; Vinyals et al. 2015; Chen and Zitnick 2015; Fang et al. 2015; Donahue et al. 2015). As with image classification, much of the progress has driven by the ability to train deep learning models at scale as well as the availability of large, diverse vision and language datasets, such as the COCO dataset (Lin et al. 2014; Chen et al. 2015).

1.1 Interpretation

Describing images with natural language captions is the problem of interpretation or understanding: one needs to analyze and interpret what the semantics / meaning of a given input scene are. In the spirit of interpreting what exists in the image, one can either just attempt to convey the semantic essence of the image based on learning from human descriptions provided at training time, or take context into account and convey a more context-aware interpretation of the scene, which takes into account some context in which we wish to describe the scene of interest. The idea of taking context into account when describing language is related to the notion of pragmatics, which is a branch of linguistics which studies how context affects the meaning of words. Below I discuss my work on tackling each of these sub-problems in interpretation of visual scenes.

1.1.1 Evaluating for Human-like Descriptions

While image captioning models are trained with the maximum-likelihood objective at training time, it is unclear how to evaluate image captions in the test scenario. While an obvious choice is the perplexity metric which is closely related to the maximum-likelihood

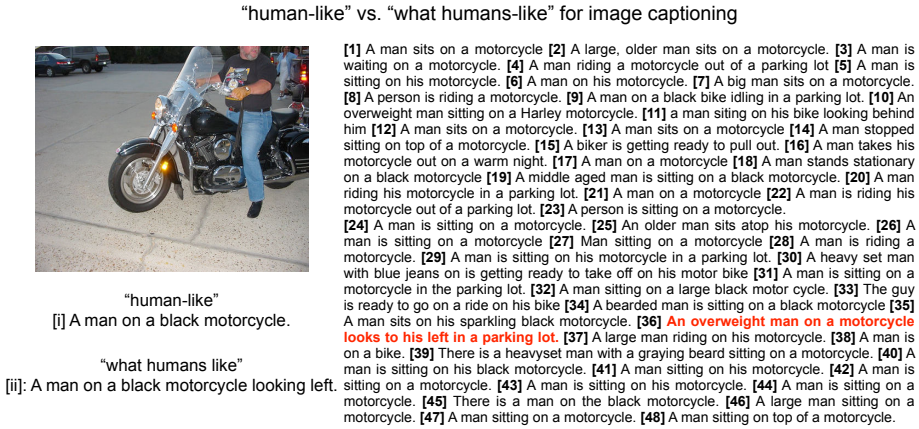


Figure 1.1: Illustration of tradeoffs for evaluating image captioning. I show an input image (top) and two captions "man is sitting on a motorcycle" and "man is sitting on a motorcycle looking left" on the left, and 48 captions written by humans for the image on the right. Interestingly, the caption which is preferred by humans mentions the fact that the person is "looking to the left" which is only present in one out of 48 human sentences. In general, my study indicates that humans have a preference for a more detailed caption such as the second one. However, most human captions are not as detailed (in fact in this image many people might not even notice that the person is looking left when writing a caption, see right). This makes evaluation for what "humans-like" tricky, since by ground-truth is not available for it. Instead, it is more tractable to evaluate for a sentence that is human-like, for which we ground-truth is sufficient for evaluation, almost by definition. Thus, my work focuses on building an evaluation protocol for finding the caption that captures the consensus in human-like descriptions.

training objective, previous work (Kulkarni et al. 2011; Vinyals et al. 2015; Liu et al. 2017) has found that it is not well correlated with human judgments of caption quality. Given this ambiguity in choices for evaluation, I set out to systematically understand, firstly, what it means to generate a good image caption, and then attempted to operationalize this into an evaluation metric. Our empirical findings suggest that it is hard to evaluate for the quality of a caption, but it is more tractable to evaluate captions based on whether they are human-like (see Fig. 1.1). Thus, in Chapter. 4 I propose a new automatic consensus metric of image description quality CIDEr (Consensus-based Image Description Evaluation). This metric measures the similarity of a generated sentence against a set of ground truth sentences written by humans, and shows high agreement with human judgments on human-likeness of captions.

1.1.2 Generating Context-aware Image Descriptions

While it is desirable to caption an image with a natural language description, it might not always be what an agent needs to accomplish in say, an interaction with a human. Often in conversation and discussions we wish to be pragmatic, take context into account emphasize or talk about selected aspects. For instance, given two images say one of a passenger jet (target image) and another of a propeller jet (distractor image) (Fig. 1.2), it is dissatisfying to caption the passenger jet image with "an aircraft flying through the sky" if the task is to distinguish the target image from the distractor. In contrast, a more human-like thing would be to say "a passenger jet flying through the sky", to refer to the passenger jet image. One approach for this would be to collect training data of language used in context, for example, discriminative ground truth utterances from people describing images in context of other images. Unfortunately, collecting such data has a prohibitive cost, since the space of objects in possible contexts is often too large. Furthermore, in some cases the context in which we wish to be pragmatic may be unknown apriori. For example, a free-form conversation agent may have to respond in a context-aware or discriminative fashion depending upon the history

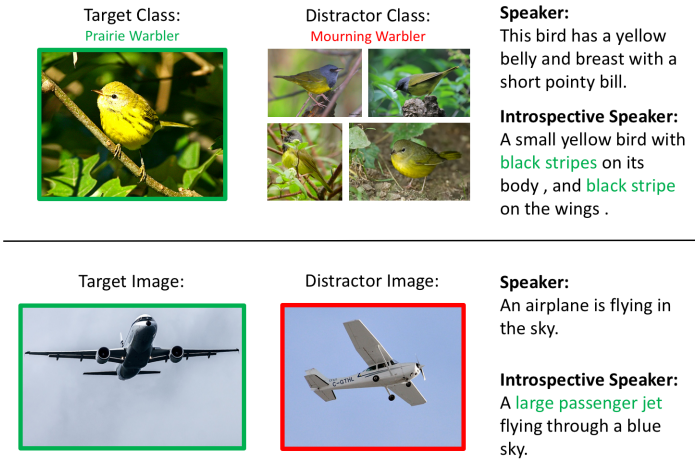


Figure 1.2: Illustration of situations where it is beneficial to model context when generating image captions: 1) *justification*: Given an image of a bird, a target (ground-truth) class (green), and a distractor class (red), one wants to describe the target image to explain why it belongs to the target class, and not the distractor class. Images from the distractor class are only shown for illustration, and are not provided to the algorithm. 2) *discriminative image captioning*: Given two similar images, one wants to produce a sentence to identify a target image (green) from the distractor image (red). "Speaker" refers to outputs from a standard image captioning model, while "Introspective Speaker" is my proposed approach which takes context into account to generate a more situation-relevant caption.

of a conversation. Thus, in Sec. 4.2, I study the problem of generating such context-aware image captions given access to just context agnostic (regular image captioning) data at training time. The core contribution of this work is in devising a novel inference algorithm to perform efficient search for context-aware captions. Our results suggest the approach offers consistent improvements over baseline "regular" image captioning models as well as previous approaches for incorporating context (Andreas and Klein 2016).

1.2 Grounding

The symbol grounding problem (Harnad 1990) is a core problem in cognitive science and artificial intelligence and is concerned with how manipulation of abstract symbols alone might not constitute truly intelligent behavior. The core idea is that words or symbols on a piece of paper or in the bits of a digital computer have no intrinsic meaning of their own. This can be understood intuitively by considering what happens when a person who does not know chinese reads chinese characters on a sheet of paper – the symbols are not grounded for this person and thus he understands chinese no better than a stream of random characters. Clearly, manipulating symbols need not correspond to an understanding of the meaning of the concepts the symbols represent, which seems central to intelligent behavior. Thus, for symbols to acquire meaning it seems necessary to understand them in a grounded context, taking into account what the symbol refers to in the physical world.

Not only is a grounded understanding of concepts important for AI in general, grounding concepts into the physical world shared by us humans can help derive inferences from machine learning models which are more human-like. I describe below my line of work on using grounding for modeling common sense, and in learning "grounded" word embeddings.

1.2.1 Learning Common Sense

Def:[1] Common sense is a basic ability to perceive, understand, and judge things, which is shared by ("common to") nearly all people and can reasonably be expected of nearly all people without any need for debate.

We see that almost by definition, an intelligent artificial agent should be able to behave in a manner that indicates understanding of commonsense knowledge. While characterizing what consistutes commonsense knowledge in general, and learning common sense are grand challenges in artificial intelligence, in this thesis we will consider the specific question of judging or figuring out if a concept "squirrel looks at nuts" happens in the real world or not. Most humans would agree that this concept is plausible. How do we design intelligent agents which are able to make such inferences? This task is trickier than it seems on the surface. One could imagine mining large amounts of text occuring on the web to see if the concept "squirrel looks at nuts" has been written about in natural language. While some commonsense knowledge is explicitly stated in human-generated text and can be learnt by mining the web, much of it is unwritten. It is often unnecessary and even unnatural to write about commonsense facts[2].

Let us consider now how a grounding based solution might be constructed for this problem. Consider that we have seen text describing that the "squirrel wants nuts", which admittedly is a more interesting thing to talk about and can conceivably be found in human-written natural language. Let us now imagine how "squirrel wants nuts" might look like. It is easy to realize that when we think about the visual depiction of someone "wanting" something, they will likely also be "looking at" the item they "want" (see Fig. 1.3). Thus by considering the perceptual grounding of "looking at" into the physical world, we can make an inference that "squirrel looking at nuts" is a plausible assertion about our world.

[1] https://www.wikipedia.org/

[2] This is known as the problem of reporting bias in text. For reference, If the frequency of mention was an indication of occurrence in the real world, people are 3 times more likely to be murdered than they are to inhale, and people inhale 6 times as often as they exhale (Gordon and Van Durme 2013)

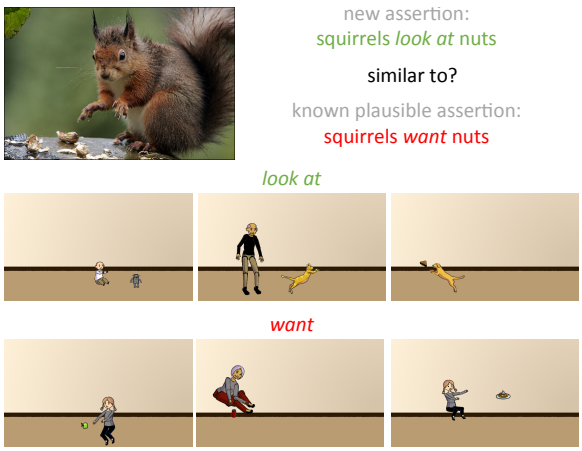


Figure 1.3: I consider the task of assessing how plausible a commonsense assertion is based on how similar it is to known plausible assertions. I argue that this similarity should be computed not just based on the text in the assertion, but also based on the visual grounding of the assertion. While "wants" and "looks at" are semantically different, their visual groundings tend to be similar. I use abstract scenes made from clipart to provide the visual grounding. These abstract scenes are completely annotated, and thus provide us access to rich semantic features which allows us to ground complicated relations such as "wants" and "looks at" into vision, which is harder to do using real images.

Just understanding that grounding is a part of the solution is not sufficient for modeling commonsense knowledge. Our second key insight is that while visual common sense is depicted in visual content, it is the semantic features that are relevant and not low-level pixel information. In other words, photorealism is not necessary to learn common sense. In Sec. 5.1 I explore the use of human generated abstract scenes made from clipart for learning common sense. In particular, I reason about the plausibility of an interaction or relation between a pair of nouns by measuring the similarity of the relation and nouns with other relations and nouns we have seen in abstract scenes. Following the grounding argument above, my work not only computes this similarity in text, but also learns an alignment function between text and vision to use vision to compute this similarity. I show that the commonsense knowledge we learn is complementary to what can be learnt from sources of text.

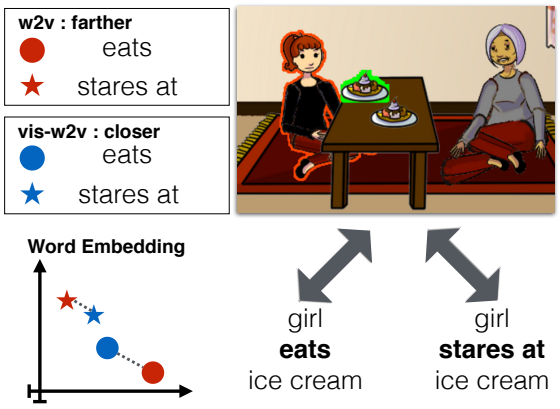


Figure 1.4: My work grounds text-based word2vec (w2v) embeddings into vision to capture a complimentary notion of visual relatedness. My method (vis-w2v) learns to predict the visual grounding as context for a given word. Although eats and stares at seem unrelated in text, they share semantics visually. Eating involves staring or looking at the food that is being eaten. As training proceeds, embeddings change from w2v (red) to vis-w2v (blue).

1.2.2 Visual-Word2vec

Word embeddings (Mikolov et al. 2013; Pennington, Socher, and Manning 2014) are continuous valued vector representations for words typically learnt to capture distributional similarity of words. That is, words which occur in the same context tend to have similar representations. These vector representations for words, trained on large datasets such as wikipedia, or the google news corpus are often used as generic features for words. Inspired by the results on modeling commonsense using grounding, I next focus on the problem of learning such word embeddings which are grounded in abstract scenes. Xu *et.al* (Xu et al. 2014a) and Lazaridou *et.al* (Lazaridou, Pham, and Baroni 2015) are some of the early works which focus on learning such grounded word representations. However, while these approaches ground words into generic "real" image features to capture appearance based similarity (to capture that say, "cat" and "dog") are similar concepts, we are interested in learning word embeddings which capture high-level notions of visual similarity such as realizing that the concepts "eats" and "stares at" are more similar than what purely distributional signals might indicate (Fig. 1.4). Similar to my previous work, we accomplish this by grounding words into abstract scenes made of clipart objects. An embedding based

approach, such as the one explored here, is more general than the alignment based approach in the previous paragraph since it allows us to use the model for inference even in contexts where the image to "align" to might not be available (*e.g.* in text-based image retrieval).

1.2.3 Sound-word2vec

One can also consider grounding word embeddings in other relevant modalities such as sound. Some of my more recent work (Vijayakumar, Vedantam, and Parikh 2017), proposes sound-word2vec - a new embedding scheme that learns specialized word embeddings grounded in sounds. For example, we learn that two seemingly (semantically) unrelated concepts, such as leaves and paper are similar due to the similar rustling sounds they make. Our embeddings prove useful in textual tasks requiring aural reasoning including text-based sound retrieval and discovering foley sound effects used in movies.

1.2.4 Grad-CAM

While my previous grounding works reason about the grounding for an observed concept in text, one can also ask for the grounding for the predictions from a visual recognition model. Similar to the arguments for grounding with generic symbols, one can argue that in order to achieve understanding of images, a model must be able to ground its predictions into the input image, otherwise one cannot say that the model has truly "understood" the image any more than a symbol manipulation machine has understood the grounding of a concept. In work led by Ramprasaath Selvaraju (Selvaraju et al. 2017), we addresses this problem of providing explanations for predictions from deep learning models operating on images as input, by grounding the predictions back into the input image in the form of a spatial heat map. For example, given an image of a cat and a dog, Grad-CAM is able to localize the region of the image for a concept "cat" when asked for an explanation. This work closes the loop between interpretation and grounding, by providing grounding for interpreted outputs, just as humans would be able to explain the supporting evidence for why they think an

image contains a cat when classifying an image.

1.3 Imagination

In semantics, one studies two different kinds of meanings for words: *intension* (Fox and Lappin 2005) and *extension* (Fox and Lappin 2005). The notion of *intension* (Dennett 1983) talks about how certain "things, events or states in the world have the interesting property of being about certain other things, events or states.". Applied to symbols in language, intension basically refers to the grounding of the symbol to some concept, or referent (Searle 1980). In addition to intension, one can also talk about the *extension* of concepts, which identifies the range of applicability of a concept. For example, the intension of the concept "car" is that is powered by an engine, has four wheels, has a steering *etc*, while the extension of the concept "car" is the set of all possible cars, each of which satisfies the intension of the concept (by definition).

In the imagination chapter, I aim to build models which can ground concepts into perception while capturing the intension and extension of the concept. That is, given a concept, one would like to generate images which denote that concept (capture the intension) and are sufficiently diverse (span the natural variation of the concept). In more technical terms, this is the problem of building generative models of the pixels of an image conditioning on a concept specification. For the running example of a "car", one would expect to see images of cars when specifying the concept "car", which captures intension while one would also expect to see say, images of "sportscars", "sedans" and "hatchbacks" which are all different kinds of cars, when we ask the model to generate multiple images (this captures extension). I pursuit of this goal, I will build on top of a previous approaches for generative image modeling called the Variational Autoencoder (Kingma and Welling 2014a), and extend it to joint models of multiple modalities.

In particular I study joint models of images and concept descriptions, where we represent concept descriptions in terms of a fixed length vector of discrete attributes. This allows us to

specify an exponentially large set of concepts using a compact, combinatorial representation. By specifying different subsets of attributes, we can generate concepts at different levels of granularity or abstraction and measure their intension and extension. In particular, one can characterize the kind of interactions we look to model as intersectional modification, where specification of more attributes leads to a more specific concept, denoted by the intersection of the original attributes. In general, words do not always compose as intersections of the entailed concepts, for example, the set of "fake guns" is not the intersection of all "fake items" and "guns". While handling such combinations remains an open challenge, in this work we will place our focus on building inductive biases into our models for intersectional modification. Overall, in the imagination chapter, I make contributions in terms of modeling, choice of objective and evaluation metrics for this task.

1.4 Levels of Analysis

David Marr's levels of analysis state that we should think about artificial intelligence at three different levels of abstraction (Kitcher 1988); namely, computational, algorithmic, and implementation. In the current context, one way to understand this framework is to think of computational considerations as specifying input output mappings between different modalities of interest, algorithmic considerations as being concerned with how to operationalize such mappings, while implementation is concerned with how intelligent behavior is implemented (say in the brain). With this scaffolding, one can understand the contributions from this thesis as pertaining to both the computational and algorithmic levels. At the computational level, we will study novel and intuitive input output mappings and problem setups which can better help model human-like inferences. On the other hand a different line of this work will build algorithmic tools and propose novel objectives which lead to intuitive, human-like inferences. This view presents another stratification of the contributions from this thesis apart from the one based on interpretation, grounding and imagination. In the next section, we will discuss some background material which is

important to place in context the algorithmic contributions from this thesis, followed by a discussion of the related work in the chapter after that.

CHAPTER 2

BACKGROUND

We will first cover some necessary background material which will be useful to understand and put the algorithmic contributions from this thesis in context. Specifically, we will talk about models for image captioning (Vinyals et al. 2015) and a class of generative models called variational autoencoders (Kingma and Welling 2014a). If you are familiar with these topics, you can skip the background part of this chapter and move to the related work.

2.1 Background: Neural Image Captioning

A strawman approach to neural image captioning typically consists of two parts: 1) an image encoder, which is usually a deep convolutional neural network (LeCun et al. 1998) applied to an input image $\mathbf{I}$ and 2) a probabilistic language decoder (Bengio et al. 2003) which is typically parameterized by a recurrent neural network, which generates a natural language description $\mathbf{s}$ (see Fig. 2.1).

Typical approaches train such an encoder-decoder model using the following maximum likelihood objective, shown here for a single sample (more recent works also explore other objectives (Liu et al. 2017; Ren et al. 2017)):

$$\mathcal{L}(\theta) = \arg\max \log p_\theta(\mathbf{s}|\mathbf{I}) \tag{2.1}$$

where, θ are the parameters of the model that we would like to estimate. In general, sentences can be of unbounded length, we will assume the following factorization for the joint distribution over all the words in a sentence $\mathbf{s} : \{\mathbf{s}_t\}$:

$$\log p(\mathbf{s}|\mathbf{I}) = \sum_{t=1}^{T} \log p(\mathbf{s}_t|\mathbf{I}, \mathbf{s}_0, \cdots, \mathbf{s}_{t-1}) \tag{2.2}$$

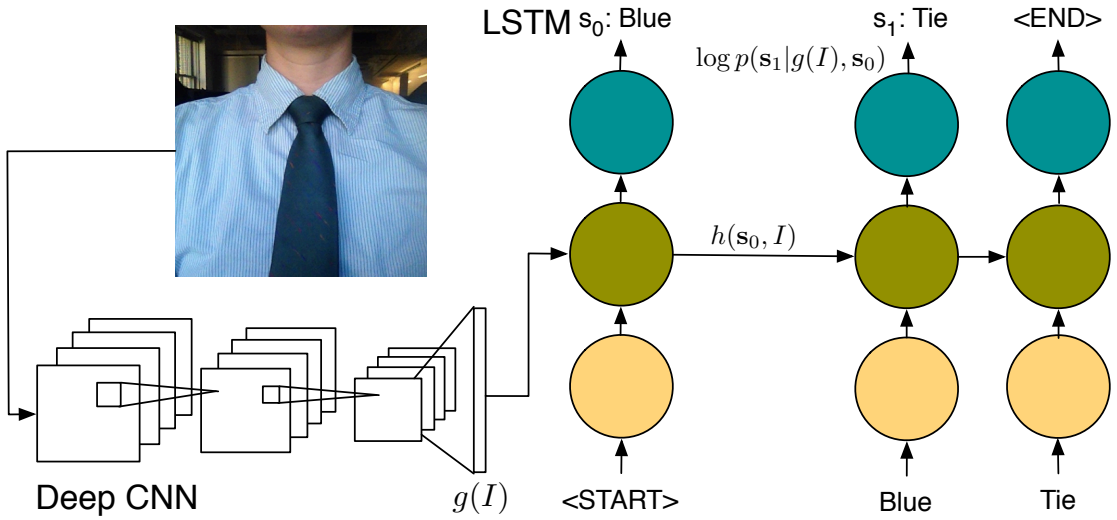


Figure 2.1: A basic sketch of a deep image captioning model. The input image $\mathbf{I}$ is passed through a Deep CNN to extract the representation $g(\mathbf{I})$, which gets fed to an LSTM to generate a caption. The LSTM is the language model which gets trained using maximum-likelihood.

We can be more explicit about the function $g(\mathbf{I})$ computed by the convolutional neural network, and write the above equation as:

$$\log p(\mathbf{s}|\mathbf{I}) = \sum_{t=1}^{T} \log p(\mathbf{s}_t|g(\mathbf{I}), \mathbf{s}_0, \cdots, \mathbf{s}_{t-1})$$

We will discuss more details of each of the components of this image captioning architecture below:

Image Encoder: Convolutional Neural Networks

A convolutional neural network (CNN) (LeCun et al. 1998) is a specific inductive bias on the form of the mapping $g(\mathbf{I}) : \mathbf{I} \rightarrow \mathbb{R}^D$, which takes into account that the mapping we learn to extract representations from images should be translation invariant. Intuitively, this means that the same filter should be able to pick out a cat regardless of where it occurs spatially in the image. More concretely, we process the input image with a "filter", by placing it at different spatial locations and obtain the responses for each location, where

the parameters of the filter stay the same regardless of which spatial location we apply the filter. We can have multiple filters in each layer, such that the end result is like applying a filter bank to the inputs at a given layer in the network. An attractive feature of deep CNNs is that with multiple layers of non-linearity, individual neurons in the network implicitly learn semantically meaningful concepts ranging from simple textures and shapes to whole or partial objects forming a dictionary of concepts. This compositionality is the second inductive bias that the deep convolutional neural network have because of depth. Together, modeling translation invariance and compositionality and training the model end-to-end using the backpropagation algorithm and stochastic gradient descent has been the recipe for much of the fundamental progress in computer vision in recent years.

Language Decoder: Recurrent Neural Networks

Now that we have a representation $g(\mathbf{I})$ for the image $\mathbf{I}$, let us look at how one would go about modeling the output distribution over sentences $p(\mathbf{s}|\mathbf{I})$. From Eqn. 2.2, let us assume we have access to some joint representation of the previous words produced till timestep t and the input image $\mathbf{I}$, given by $h_{t-1}(\mathbf{I}, \mathbf{s}_0, \cdots, \mathbf{s}_{t-1})$. Then, we can write the maximum-likelihood objective problem at a timestep t as follows:

$$\log p(\mathbf{s}_t|g(\mathbf{I}), \mathbf{s}_0, \cdots, \mathbf{s}_{t-1}) = \log p(\mathbf{s}_t|h_{t-1}(\mathbf{I}, \mathbf{s}_0, \cdots, \mathbf{s}_{t-1}), \mathbf{s}_{t-1})$$

where, h denotes the history, $\mathbf{s}_{t-1}$ is the ground truth token at the previous timestep which is fed as input at timestep t, and $\mathbf{s_t}$ is the ground truth token at the current timestep. This mapping is implemented using a long-short term memory (LSTM) recurrent neural network (Hochreiter and Schmidhuber 1997). Denoting σ as the sigmoid function $\frac{\exp^a}{1+\exp^a}$, $\cdot$ as the dot product between vectors, and $\odot$ as the element-wise vector product, $[\cdot]$ as vector

concatenation, we can describe the LSTM with the following equations:

$$
\begin{aligned}
i_t &= \sigma(W_i \cdot [\mathbf{s}_{t-1}, \mathbf{h}_{t-1}]) \\
f_t &= \sigma(W_f \cdot [\mathbf{s}_{t-1}, \mathbf{h}_{t-1}]) \\
o_t &= \sigma(W_o \cdot [\mathbf{s}_{t-1}, \mathbf{h}_{t-1}]) \\
c_t &= f_t \odot c_{t-1} + i_t \odot tanh(W_c \cdot [\mathbf{s}_{t-1}, \mathbf{h}_{t-1}]) \\
h_t &= o_t \odot c_t \\
p_t(\mathbf{s}_t | g(\mathbf{I}), \mathbf{s}_0, \cdots, \mathbf{s}_{t-1}) &= softmax(\mathbf{h}_t)
\end{aligned}
$$

While the above equations look quite complicated, they are essentially trying to model the dynamics of the mapping p_t using a neural network, with parameter sharing across multiple timesteps (notice how (W_i, W_f, W_o, W_c) are shared across timesteps in the equations above). A naive approach to implementing this parameter sharing leads to instabilities when training the model with stochastic gradient descent, where gradients would either explode or vanish (Hochreiter and Schmidhuber 1997). Consequently, the LSTM architecture was proposed by Hochreiter and Schmidhuber 1997 to mitigate some of these problems. While I provide the details of the LSTM architecture here for completeness, we will not be making any changes to this architecture and will be using this model as-is during the rest of the expositions in this proposal.

Inference: Beam Search

After training the model with the maximum likelihood objective above, at inference time we must solve the following optimization problem to search for the most probable sentence as per the model:

$$\arg\max_{\mathbf{s}} \log p(\mathbf{s}|\mathbf{I})$$

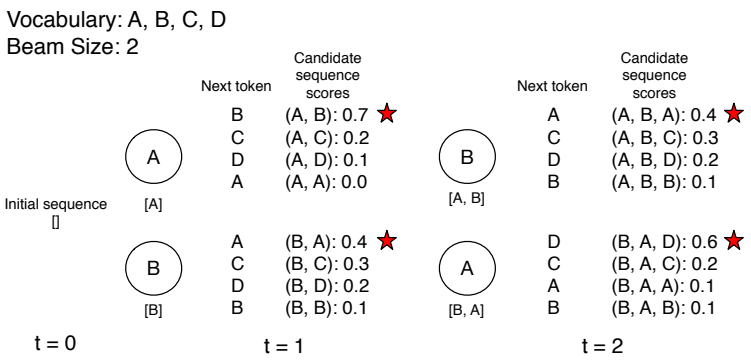


Figure 2.2: An illustration of beam search for beam size 2, and vocabulary of (A, B, C, D), setting up a simple sequence prediction problem. At each timestep, we sample all possible completions of the two beams, which for the current two beam, four token case is 8 possible completions. We pick the best scoring sequences among the 8 cases, and extend those beams into the next timestep. We continue this process until an "end" token is reached or a maximum number of timesteps have passed. The candidate sequence score in the case of image captioning, at time t is chosen to be $\log p(\mathbf{s}_0, \cdots, \mathbf{s}_t|\mathbf{I})$.

This search is intractable in general, since the set of all sentences of length T for a vocabulary $\mathcal{V}$ grows exponentially ($|\mathcal{V}|^T$). One can adopt a greedy approach optimizing for a lower bound on the true inference objective:

$$\begin{aligned}\max_{\mathbf{s}} \log p(\mathbf{s}|\mathbf{I}) &= \max_{\mathbf{s}} \sum_{t=1}^{T} \log p(\mathbf{s}_t|\mathbf{I}, \mathbf{s}_0, \cdots, \mathbf{s}_{t-1}) \\ &\geq \sum_{t=1}^{T} \max_{\mathbf{s}_t} \log p(\mathbf{s}_t|\mathbf{I}, \mathbf{s}_0, \cdots, \mathbf{s}_{t-1})\end{aligned}$$

Instead of doing this greedy maximization at every timestep, one can often do better in practice by keeping a list of top-B most promising hypotheses at every timestep (as opposed to greedy which just keeps the top-1 hypothesis). See Fig. 2.2 for a simple illustration of beam search for a beam size of 2, and a vocabulary of 4 items (A, B, C, D).

2.2 Background: Variational Autoencoder

A variational autoencoder (Kingma and Welling 2014a) is a latent variable model, say for observed images $\mathbf{x}$ with a gaussian latent variable $\mathbf{z}$. Typically we assume the following

generative process for the model: we pick a latent $\mathbf{z} \sim p(\mathbf{z})$, and then given the sampled latent, we pass it through the generative model $p(\mathbf{x}|\mathbf{z})$ to generate an image. Given this generative process, one would also like to be able to do inference given an input image, i.e. estimate its corresponding latent variable. Unfortunately, this tends to be intractable as computing $p(\mathbf{z}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{\sum_z p(\mathbf{x}|\mathbf{z})p(\mathbf{z})}$ is not always possible. Variational inference (Jordan et al. 1999) casts the problem of inference as optimization by assuming access to an approximating function $q(\mathbf{z}|\mathbf{x})$, which tries to approximate the true (intractable) distribution $p(\mathbf{z}|\mathbf{x})$. Variational inference applied to our graphical model results in the following lower bound on the true data likelihood:

$$\log p(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z}\sim q(\mathbf{z}|\mathbf{x})} \left[\log p(\mathbf{x}|\mathbf{z})\right] - \mathbf{KL}\left[q(\mathbf{z}|\mathbf{x})||p(\mathbf{z})\right] \tag{2.3}$$

This lower bound is known as the evidence lower bound or elbo. In terminology that will be useful in later chapters, we can denote this bound for $\mathbf{x}$ as $\mathrm{elbo}(\mathbf{x})$. This objective is hard to optimize using stochastic gradient descent in general because the gradient of the first term tends to have high variance because of sampling in the expectation (Kingma and Welling 2014a). The variational autoencoder derives a lower-variance estimator for the gaussian case reparameterizing samples from a gaussian as $x = \mu + \sigma \cdot \epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$. This separates out the parameters estimated by the network $q(z|x)$ from the stochasticity due to sampling, meaning that one can get low-variance unbiased gradients for optimizing Eqn. 2.3. This variational lower-bound will be the cornerstone of our approach for visual imagination, and I will show how to extend this objective to capture intension and extension of semantic concepts.

CHAPTER 3
SITUATING THE WORK

I first discuss related work on interpretation, then grounding followed by imagination. In interpretation, we will cover prior work on generating image captions, considerations for evaluation of image captions, and reasoning about pragmatics and performing context aware captioning. In grounding we will cover related work on reasoning about common sense, learning with visual abstraction, and learning grounded word embeddings. Finally we will cover related work in the space of "imagination".

3.1 Interpretation

3.1.1 Image Caption Generation

Various methods have been explored for generating full descriptions for images. Traditionally, the techniques have either been based on retrieval (Farhadi et al. 2010; Ordonez, Kulkarni, and Berg 2011; Hodosh, Young, and Hockenmaier 2013) or generation (Mitchell, Han, and Hayes 2012; Kulkarni et al. 2011; Yatskar et al. 2014; Rohrbach et al. 2013). Approaches which perform retrieval cannot create novel sentences – they attempt to copy them from a database of sentences which have already been written, while generation based approaches try to put together a sentence from scratch given an input image.

While some retrieval-based approaches use global retrieval (Farhadi et al. 2010), others retrieve text phrases and stitch them together in an approach inspired by extractive summarization (Ordonez, Kulkarni, and Berg 2011). The recent wave of progress in image description models stems from deep neural network approaches trained end-to-end on the task of generating sentences from pixels by combining Convolutional and Recurrent Neural Networks (Karpathy and Fei-Fei 2015; Chen and Zitnick 2015; Donahue et al. 2015; Vinyals

et al. 2015; Mao et al. 2015a). Beyond this line of work, more recently we have seen advances in terms of using image attention for caption generation, where the model learns a categorical latent variable representing the importance of image regions (Xu et al. 2015; Lu et al. 2017), approaches which use object level attention for image captions (Anderson et al. 2017) and approaches optimizing higher level task driven metrics (see below section for a discussion of metrics) as opposed to using maximum-likelihood estimation (Liu et al. 2017; Ren et al. 2017).

3.1.2 Evaluating Image Captioning

Metrics: While it is exciting to see all the progress being made on the modeling side for image captioning, it is crucial to establish appropriate evaluation protocols to measure and benchmark "real" progress in the long run, as has been seen in various tasks in computer vision, such as detection (Everingham et al. 2015; Deng et al. 2009), segmentation (Everingham et al. 2015; Martin et al. 2001), and stereo (Scharstein and Szeliski 2002).

When constructing evaluation protocols for high-level tasks such as image captioning, it is natural to consider evaluating by collecting human judgments for the "quality" of the generated outputs (Vinyals et al. 2015). However, human studies are expensive, hard to reproduce and slow to evaluate which makes them impractical. Thus, automated metrics are commonly desired.

A automated metric for image captioning would typically have an API which takes as input a generated caption, and set of ground truth human-written captions outputs some measure of similarity between them.

To be useful in practice, scores from such automated metrics should agree well with human judgment. Some popular metrics used for image description evaluation at the time of my work were BLEU (Papineni et al. 2002) (precision-based) from the machine translation community and ROUGE (Lin 2004) (recall-based) from the summarization community. Unfortunately, these metrics have been shown to correlate weakly with human

judgment (Kulkarni et al. 2011; Elliott and Keller 2014; Callison-burch and Osborne 2006; Hodosh, Young, and Hockenmaier 2013). For the task of judging the overall quality of a description, the METEOR (Elliott and Keller 2014) metric has shown better correlation with human subjects. Other proposed metrics rely on the ranking of captions (Hodosh, Young, and Hockenmaier 2013) and cannot evaluate novel image descriptions. In Chapter. 4 I describe my work on a consensus-based protocol for image description evaluation which captures the consensus in human written captions and uses it for evaluation. The CIDEr metric we propose outperforms previous metrics when it comes to matching human judgments of quality. More recently, Anderson et al. 2016 proposed the SPICE (Semantic Propositional Image Captioning Evaluation) for evaluating image captioning, which matches sentences by first extracting a semantic parse of the sentences and then computing similarity based on the corresponding parses using a graph based similarity measure. In general, the SPICE metric captures human judgments of caption quality really well (Anderson et al. 2016), and is the only approach which consistently ranks human captions better than machine generated ones (Anderson et al. 2016). However, as noted by (Liu et al. 2017), SPICE ignores syntactic quality, which basically means that a high SPICE score is not sufficient to declare that a sentence is good, since one could construct a sentence with repetitive phrase structures which has a perfect semantic parse. Consequently, Liu et al. 2017 propose to use a combination of the SPICE and CIDEr metrics (which they imaginatively call SPIDEr) to evaluate image captioning approaches, which might as well be the way forward in image captioning evaluation space.

3.1.3 Pragmatics and Context-aware Image Captioning

While capturing the semantic essence of an image and expressing it in natural language is useful, it is ultimately ill posed, since in many scenarios, context plays a major role in deciding what is relevant to say about an image. This notion of accounting for context when generating utterances is related to the topic of pragmatics, which studies how context

influences the usage of language.

Early work on pragmatics stems from Grice 1975 who analyzed how cooperative multi-agent linguistic agents could model each others' behavior to achieve a common objective. Consequently, a lot of pragmatics literature has studied higher-level behavior in agents including conversational implicature (Benotti and Traum 2009) and the Gricean maxims (Vogel et al. 2013).

These works aim to derive pragmatic behavior given minimal assumptions on individual agents and typically use hand-tuned lexicons and rules. More recently, there have been exciting developments on applying reinforcement learning (RL) techniques to these problems (Mordatch and Abbeel 2017; Das et al. 2017; Lazaridou, Peysakhovich, and Baroni 2016), requiring less manual tuning.

In Sec. 4.2 we are also interested in accounting for context and pragmatics, but are interested in the specific case of captioning images by taking pragmatics into account. Other works model ideas from pragmatics to learn language via games played online (Wang, Liang, and Manning 2016) or for human-robot collaboration (Tellex et al. 2014). In a similar spirit, in Sec. 4.2 we are interested in applying ideas from pragmatics to build systems that can provide justifications by explaining why an image contains a given category as opposed to a different (but visually similar) category, and provide discriminative image captions by taking into account distractor images as context in image captioning models.

Most relevant to our work is the recent work on deriving pragmatic behavior in abstract scenes made with clipart, by Andreas and Klein 2016. The approach of Andreas, and Klein first trains a regular image captioning model, and then reranks the captions sampled from this model on the basis of how class discriminative they might be by training a ranking function on aligned pairs of sentences and classes (which they call the listener model). Unlike their technique, our proposed approach does not require training a second listener model and supports more efficient inference. See Sec. 4.2 for more details.

Moreover, ours is not the first work to study how to account for context when generating

image captions. Sadovnik et al. 2012 first studied a discriminative image description task, with the goal of distinguishing one image from a set of images. Their approach incorporates cues such as discriminability and saliency, and uses hand-designed rules for constructing sentences. In contrast, we develop inference techniques to induce discriminative behavior in neural models. The reference game from Andreas and Klein 2016 can also be seen as a discriminative image captioning task on abstract scenes made from clipart, while we are interested in the domain of real images. The work on generating referring expressions by Mao et al. 2015b generates discriminative captions which refer to particular objects in an image given context-aware supervision. Our work is different in the sense that we address an instance of pragmatic reasoning in the common case where context-dependent data is not available for training.

3.2 Grounding and Commonsense Reasoning

I next discuss some of the related work in the literature on the lines of modeling commonsense knowledge and grounding symbols into perceptual cues. I also discuss approaches which make use of abstract scenes created with clipart, which forms our primary visual grounding modality in this line of work.

3.2.1 Modeling Commonsense Knowledge

Common Sense and Text

A sensible first approach to modeling commonsense knowledge would be to search for them in knowledge bases built from text. There is a rich line of works which learn relations between entities to build such knowledge bases either using machine reading (e.g., Knowledge Vault (Dong et al. 2014), NELL (Carlson et al. 2010), ReVerb (Etzioni et al. 2011)) or using collaboration within a community of users (e.g., Freebase (Bollacker et al. 2008a), Wikipedia[1]).

[1] http://www.wikipedia.org/

Common Sense and Vision

Since text suffers from a reporting bias (Chapter. 1), it is interesting to consider the grounding for textual concepts when reasoning about visual commonsense knowledge. In this line of work, my work Sec. 5.1 and concurrent work on VisKE (Sadeghi, Divvala, and Farhadi 2015) study the same task of evaluating the plausibility of commonsense assertions using visual cues. In VisKE, the visual cues are derived from webly-supervised detection (Divvala, Farhadi, and Guestrin 2014) models, while we use abstract scenes and text embeddings. Our goal is to explore if one can make human-like inferences about plausibility of assertions simply by analyzing abstract scenes made of clipart, by passing the intermediate hard problem of recognizing various entities and their relations in real images. Building scalable detectors for a large variety of objects and relations is still an open research problem in vision (Li et al. 2017; Lu et al. 2016).

Traditionally, a popular use of commonsense knowledge in vision has been for modeling context for improved recognition (Divvala et al. 2009; Fouhey et al. 2012). Recently, there has been a surge in interest in high-level "beyond recognition" tasks which can benefit from external knowledge beyond what is depicted in the image (Berg et al. 2012; Hays and Efros 2008; Khosla et al. 2014; Pickup et al. 2014; Pirsiavash, Vondrick, and Torralba 2014).

In terms of modeling relations, Zhu, Fathi, and Fei-Fei 2014 use attribute and action classification along with information from various textual knowledge bases to perform tasks like zero-shot affordance prediction for human-object interactions. While their dictionary of relations was specified manually and limited to 19 inter-object relations. My work explores a larger number of *free-form* relations (213 in total) extracted from text. Johnson et al. 2015 extract scene graph representations from images based on a recent large scale dataset of scene graphs (Krishna et al. 2016). LEVAN (Divvala, Farhadi, and Guestrin 2014) trains detectors for a variety of bigrams (e.g., jumping horse) from google n-grams using web-scale image data. NEIL (Chen, Shrivastava, and Gupta 2013) analyzes images on the web to learn visual models of objects, scenes, attributes, part-of, and other ontology relationships. Unlike

these works, the focus of my work is less on appearance models and more on the underlying semantics. Recent work has also looked at mining *semantic* affordances, *i.e.* inferring whether a given action can be performed on an object (Chao et al. 2015). In contrast, I am interested in the more general problem of predicting the plausibility of interactions or relations between pairs of objects. Lin and Parikh 2015 propose to learn visual common sense and use it to answer textual fill-in-the-blank and visual paraphrasing questions, by imagining a scene behind the text. While they model visual common sense in the context of a scene, my task is at a more atomic level – reasoning about the plausibility of a specific relation or interaction between pairs of objects.

3.2.2 Learning from Visual Abstraction

The idea of using abstract visual concepts for scene understanding first appeared in the work of Zitnick and Parikh (Zitnick and Parikh 2013; Zitnick, Vedantam, and Parikh 2016). Zitnick, Parikh, and Vanderwende 2013 learns the visual interpretation of sentences and generates scenes for a given input sentence. Fouhey and Zitnick 2014 learn the dynamics of objects in scenes from temporal sequences of abstract scenes. Antol, Zitnick, and Parikh 2014 learn models of fine-grained interactions between pairs of people using visual abstractions, and evaluate their models on real images from the web. Lin and Parikh 2015 "imagine" abstract scenes corresponding to text, and use the common sense depicted in these imagined scenes to solve textual tasks such as fill-in-the-blanks and paraphrasing. Andreas and Klein 2016 use a dataset of abstract scenes to create fine-grained distractor images for context-aware image captioning, while Wu, Tenenbaum, and Kohli 2017 show how to learn a disentangled and structured scene representation using abstract scenes. Other work which uses abstract scenes includes (Zhang et al. 2015; Antol et al. 2015; Ortiz, Wolff, and Lapata 2015; Kottur et al. 2015).

3.2.3 Learning Word Embeddings

Word embeddings are continuous valued vector representations of discrete word tokens encoded using a vocabulary. Unlike a one-hot representation for words, which places every word at the same distance, word embeddings are useful because they help reason about semantically similar words, by making the representational choice of continuous valued vector spaces, and training them to model distributional similarity (Mikolov et al. 2013). These embeddings are useful for a number of tasks in natural language processing and are popularly used as features for a number of tasks (Rocktäschel et al. 2016; Lample et al. 2016; Gao et al. 2015; Irsoy and Cardie 2014).

Grounded Word Embeddings: Given the importance of grounding symbols into the physical world (Chapter. 1), previous works have studied how to ground word embeddings into vision. In contrast to these approaches, my work (Sec. 5.2) studies grounding the word embeddings not into generic image features (which capture appearance) but into abstract scenes made with clipart (which capture fine-grained visual information). More concretely, Xu et al. 2014b and Lazaridou, Pham, and Baroni 2015 use visual cues to improve the word2vec representation by predicting real image representations from word2vec and maximizing the dot product between image features and word2vec respectively. Other works use visual and textual attributes (*e.g.*vegetable is an attribute for potato) to improve distributional models of word meaning (Silberer, Ferrari, and Lapata 2013). In contrast to these approaches, our set of visual concepts need not be explicitly specified, it is implicitly learnt in a clustering step. Apart from vision, works have also studied the problem of grounding words in sounds. While Lopopolo and Miltenburg 2015 show preliminary results on using sound to learn distributional representations, Kiela and Clark 2015 build on ideas from Bruni, Tran, and Baroni 2014 to learn word embeddings that respect both linguistic and auditory relationships by optimizing a joint objective. I have been involved in a work on grounding sounds, where we learn "specialized" embeddings that exclusively fit to relationships defined by sounds, using word2vec embeddings for smoothness. Similar

to previous findings (Melamud et al. 2016), our work finds that specialized embeddings outperform both language-only and other multi-modal embeddings on the downstream tasks of interest (we study text-based sound retrieval and foley sound discovery).

3.3 Imagination

Recent years have seen significant advances in generative image model, based on fundamental advances in modeling techniques via. the variational autoencoder (see Chapter. 2) and a different class of techniques called generative adversarial networks (GANs) (Goodfellow et al. 2014a) which are implicit likelihood estimation techniques, *i.e.* they allow us to perform estimation without explicitly computing any likelihood. Some notable extensions have also been proposed to these frameworks, to learn representations which disentangle factors of variation in images. For the variational autoencoder framework, an important technique is β-VAE (Higgins et al. 2017a), that tweaks the weightage given to the likelihood and the KL divergence terms in a regular VAE (see Chapter. 2). For the class of generative adversarial networks, an important such technique is InfoGAN (Chen et al. 2016) which maximizes the mutual information between latent codes and generated images.

These conceptual advances have led to a flurry of approaches which look at conditional generative image models (of the form $p(\mathbf{x}|\mathbf{y})$, where $\mathbf{y}$ can be input class labels (Oord et al. 2016; Kingma et al. 2014), a vector of attributes (Yan et al. 2016a), sentences (Reed et al. 2016a), or even other images (Isola et al. 2017a) and are able to generate output images based on the conditioning.

However, in visually grounded imagination, we are not simply interested in translating an input modality into images, we would like to systematically study how the distribution over images changes as we reduce the amount of information we condition upon. To do this, we propose to use attributes for generation. Attributes offer the advantage that they allow us to specify more generic concepts by specifying a subset of attributes allowing us to easily measure the extension of the concept of interest (by measuring how diversity

along unspecified axes). Dealing with missing inputs is in general easier in joint models where one can simply marginalize out what they do not observe. Thus, we are more interested in learning a shared latent space from either descriptions $\mathbf{y}$ or images $\mathbf{x}$, than in learning a conditional model $p(\mathbf{x}|\mathbf{y})$, which means we need to use a joint, symmetric, model (Chapter. 6). Some related joint generative models are the BiVCCA objective of Wang, Lee, and Livescu 2016a, JMVAE objective of Suzuki, Nakayama, and Matsuo 2017a which are instantiations of joint variational autoencoder models. We propose to use a related but different objective called TELBO to fit such joint generative models of images and attribute labels.

Joint generative modeling in the variational autoencoder framework comes with its own challenges, since the inference networks in such models, which provide an estimate of the latent space given some attribute vectors (for instance) are discriminative and thus cannot handle missing inputs. In Sec. 6.1 we propose some ways to handle the missing attribute case.

CHAPTER 4

INTERPRETATION

In this chapter, we will discuss my line of work in extracting natural language descriptions from images, which we will call the problem of interpretation.

We will first study the important problem of evaluating image captioning approaches, and suggest an evaluation protocol based on modeling human consensus. This protocol consists of three different parts: a new metric for image captioning called CIDEr, two new datasets for evaluation, namely ABSTRACT-50S and PASCAL-50S, and a new human annotation modality for capturing human judgments of consensus.

Next, we will motivate why one needs to account for context when generating image captions in specific situations, and study two relevant tasks: justification, where the task is to explain why an image contains a target class as opposed to a given distractor class, and discriminative image captioning where the task is to compose a caption that uniquely refers to a target image relative to a distractor image. It is important to note that the task is to produce such context-aware captions without access to context-aware ground truth at training time. My work will also contribute a new dataset CUB-Justify, with human explanations for why an image contains a target class as opposed to a distractor class, enabling for systematic evaluation of future justification approaches. The main contribution of the work however, will be a novel inference algorithm which explicitly takes into account context when generating sentences, and scales with a simple modification to both the tasks of interest.

While the first work CIDEr makes computational advances suggesting how to evaluate image capitoning to genrerate more human-like captions, the second work justification studies a novel algorithm to create captions which are able to take distractor images or classes into account, leading to captions which are better aligned to the context, which is the

rational behavior one would expect from an intelligent agent.

4.1 CIDEr: Consensus-based Image Description Evaluation

Evaluation of generated image captions is challenging because intuitively, one would want to measure a number of diserable properties: grammaticality, saliency (covering main aspects), correctness/truthfulness, *etc*. Using human studies, these properties may be measured, *e.g.* on separate *one* to *five* (Mitchell, Han, and Hayes 2012; Rohrbach et al. 2013; Elliott and Keller 2014) or *pairwise* scales (Yatskar et al. 2014). Unfortunately, combining these various results into one measure of sentence quality is difficult. Alternatively, other works (Kulkarni et al. 2011; Hodosh, Young, and Hockenmaier 2013) ask subjects to judge the overall quality of a sentence.

An important yet non-obvious property exists when image descriptions are judged by humans: What humans like often does not correspond to what is human-like.[1] Below I first introduce a novel consensus-based evaluation protocol, which measures the similarity of a sentence to the majority, or *consensus* of how most people describe the image (Fig. 1.1).

One realization of this evaluation protocol uses human subjects to judge sentence similarity between a candidate sentence and human-provided ground truth sentences. The question "Which of two sentences is more similar to this other sentence?" is posed to the subjects. The resulting quality score is based on how often a sentence is labeled as being *more* similar to a human-generated sentence. The relative nature of the question helps make the task objective. We encourage the reader to review how a similar protocol has been used in Tamuz et al. 2011 to capture human perception of image similarity. These annotation protocols for similarity may be understood as instantiations of 2AFC (two alternative forced choice) (Bogacz et al. 2006), a popular modality in psychophysics.

Below I propose a new automatic *consensus* metric of image description quality – CIDEr

[1]This is a subtle but important distinction. I show qualitative examples of this in the appendix. That is, the sentence that is most similar to a typical human generated description is often not judged to be the "best" description. In this section, I propose to directly measure the "human-likeness" of automatically generated sentences.

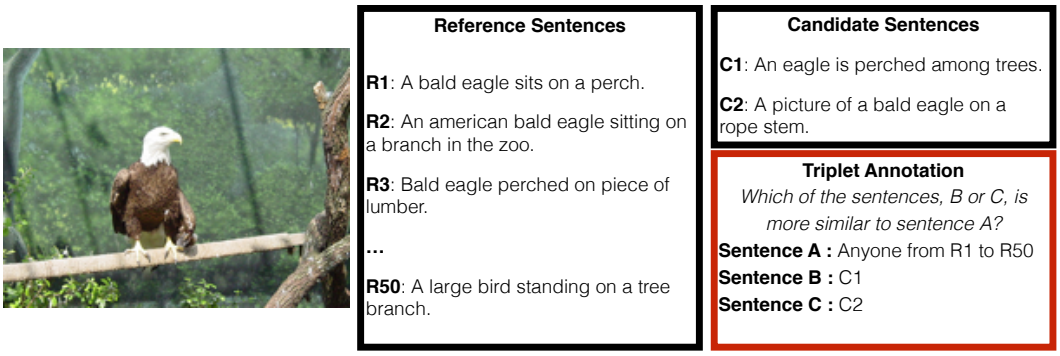
(Consensus-based Image Description Evaluation). The metric measures the similarity of a generated sentence against a set of ground truth sentences written by humans. It shows high agreement with consensus as assessed by humans. Using sentence similarity, the notions of grammaticality, saliency, importance and accuracy (precision and recall) are inherently captured by our metric. See Chapter. 3 for a discussion of how our metric compares to other metrics proposed in the literature, such as BLEU (Papineni et al. 2002), ROUGE (Lin 2004), METEOR (Lavie and Denkowski 2009) and SPICE (Anderson et al. 2016).

Existing datasets popularly used to evaluate image description approaches have a maximum of only five descriptions per image (Rashtchian et al. 2010; Hodosh, Young, and Hockenmaier 2013; Ordonez, Kulkarni, and Berg 2011). However, I find that five sentences are not sufficient for measuring how a "majority" of humans would describe an image. Thus, to accurately measure consensus, I collect two new evaluation datasets containing 50 descriptions per image – PASCAL-50S and ABSTRACT-50S. The PASCAL-50S dataset is based on the popular UIUC Pascal Sentence Dataset, which has 5 descriptions per image. This dataset has been used for both training and testing in numerous works (Mitchell, Han, and Hayes 2012; Kulkarni et al. 2011; Farhadi et al. 2010; Rohrbach et al. 2013). The ABSTRACT-50S dataset is based on the dataset of Zitnick and Parikh (Zitnick and Parikh 2013). While previous methods have only evaluated using 5 sentences, we explore the use of 1 to ~50 reference sentences. Interestingly, I find that most metrics improve in performance with more sentences.[2] Inspired by this finding, the COCO testing dataset now contains 5K images with 40 reference sentences to boost the accuracy of automatic measures (Chen et al. 2015).

4.1.1 Consensus Interface

Given an image and a collection of human generated *reference* sentences describing it, the goal of the consensus-based protocol is to measure the similarity of a *candidate* sentence

[2]Except BLEU computed on unigrams



(a) (b) (c)

Figure 4.1: Illustration of the triplet annotation modality. Given an image (a), with reference sentences (b) and a pair of candidate sentences (c, top), we match them with a reference sentence one by one to form triplets (c, bottom). Subjects are shown these 50 triplets on Amazon Mechanical Turk and asked to pick which sentence (B or C) is more similar to sentence A.

to a majority of how most people describe the image (*i.e.*the *reference* sentences). In this sub-section, I describe the human study protocol for generating ground truth consensus scores. In Sec. 4.1.5, these ground truth scores are used to evaluate several automatic metrics including the proposed CIDEr metric.

An illustration of the human study interface is shown in Fig. 4.1. Subjects are shown three sentences: A, B and C. They are asked to pick which of two sentences (B or C) is most similar to sentence A. Sentences B and C are two candidate sentences, while sentence A is a reference sentence. For each choice of B and C, we form triplets using all the reference sentences for an image. I provide no explicit concept of "similarity". Interestingly, even though I do not say that the sentences are image descriptions, some workers commented that they were imagining the scene to make the choice. The relative nature of the task – "Which of the two sentences, B or C, is more similar to A?" – helps make the assessment more objective. That is, it is easier to judge if one sentence is more similar than another to a sentence, than to provide an absolute rating from 1 to 5 of the similarity between two sentences (Bogacz et al. 2006).

I collected three human judgments for each triplet. For every triplet, I took the majority vote of the three judgments. For each pair of candidate sentences (B, C), I assign B the

winner if it is chosen as more similar by a majority of triplets, and similarly for C. These pairwise relative rankings are used to evaluate the performance of the automated metrics. That is, when automatic metrics give both sentences B and C a score, I check whether B received a higher score or C. Accuracy is computed as the proportion of candidate pairs on which humans and the automatic metric agree on which of the two sentences is the winner.

4.1.2 CIDEr Metric

My goal is to automatically evaluate for image I_i how well a candidate sentence $\mathbf{c}_i$ matches the consensus of a set of image descriptions $S_i = \{\mathbf{s}_{i1}, \ldots, \mathbf{s}_{im}\}$. All words in the sentences (both candidate and references) are first mapped to their stem or root forms. That is, "fishes", "fishing" and "fished" all get reduced to "fish." I represent each sentence using the set of n-grams present in it. An n-gram ω_k is a set of one or more ordered words. In this chapter I use n-grams containing one to four words.

Intuitively, a measure of consensus would encode how often n-grams in the candidate sentence are present in the reference sentences. Similarly, n-grams not present in the reference sentences should not be in the candidate sentence. Finally, n-grams that commonly occur across all images in the dataset should be given lower weight, since they are likely to be less informative. To encode this intuition, I perform a Term Frequency Inverse Document Frequency (TF-IDF) weighting for each n-gram (Robertson 2004). The number of times an n-gram ω_k occurs in a reference sentence $\mathbf{s}_{ij}$ is denoted by $h_k(\mathbf{s}_{ij})$ or $h_k(\mathbf{c}_i)$ for the candidate sentence $\mathbf{c}_i$. I compute the TF-IDF weighting $g_k(\mathbf{s}_{ij})$ for each n-gram ω_k using:

$$g_k(\mathbf{s}_{ij}) = \frac{h_k(\mathbf{s}_{ij})}{\sum_{\omega_l \in \Omega} h_l(\mathbf{s}_{ij})} \log \left(\frac{|I|}{\sum_{I_p \in I} \min(1, \sum_q h_k(\mathbf{s}_{pq}))} \right), \tag{4.1}$$

where Ω is the vocabulary of all n-grams and I is the set of all images in the dataset. The first term measures the TF of each n-gram ω_k, and the second term measures the rarity of ω_k using its IDF. Intuitively, TF places higher weight on n-grams that frequently occur in the reference sentence describing an image, while IDF reduces the weight of n-grams that

commonly occur across all images in the dataset. That is, the IDF provides a measure of word saliency by discounting popular words that are likely to be less visually informative. The IDF is computed using the logarithm of the number of images in the dataset $|I|$ divided by the number of images for which ω_k occurs in any of its reference sentences.

The CIDEr_n score for n-grams of length n is computed using the average cosine similarity between the candidate sentence and the reference sentences, which accounts for both precision and recall:

$$\text{CIDEr}_n(\mathbf{c}_i, S_i) = \frac{1}{m} \sum_j \frac{\boldsymbol{g^n}(\mathbf{c}_i) \cdot \boldsymbol{g^n}(\mathbf{s}_{ij})}{\|\boldsymbol{g^n}(\mathbf{c}_i)\| \|\boldsymbol{g^n}(\mathbf{s}_{ij})\|}, \tag{4.2}$$

where $\boldsymbol{g^n}(\mathbf{c}_i)$ is a vector formed by $g_k(\mathbf{c}_i)$ corresponding to all n-grams of length n and $\|\boldsymbol{g^n}(\mathbf{c}_i)\|$ is the magnitude of the vector $\boldsymbol{g^n}(\mathbf{c}_i)$. Similarly for $\boldsymbol{g^n}(\mathbf{s}_{ij})$.

I use higher order (longer) n-grams to capture grammatical properties as well as richer semantics. I combine the scores from n-grams of varying lengths as follows:

$$\text{CIDEr}(\mathbf{c}_i, S_i) = \sum_{n=1}^{N} w_n \text{CIDEr}_n(\mathbf{c}_i, S_i), \tag{4.3}$$

Empirically, I found that uniform weights $w_n = 1/N$ work the best, *i.e.* I use N = 4.

4.1.3 New Datasets

I propose two new datasets – PASCAL-50S and ABSTRACT-50S – for evaluating image caption generation methods. Both the datasets have 50 reference sentences per image for 1,000 and 500 images respectively. These are intended as "testing" datasets, crafted to enable consensus-based evaluation. For a list of training datasets, I encourage the reader to explore (Lin et al. 2014; Plummer et al. 2015; Ordonez, Kulkarni, and Berg 2011). The PASCAL-50S dataset uses all 1,000 images from the UIUC Pascal Sentence Dataset (Rashtchian et al. 2010) whereas the ABSTRACT-50S dataset uses 500 random images from the abstract scenes dataset (Zitnick and Parikh 2013). The abstract scenes

dataset contains scenes made from clipart objects.

My goal was to collect image descriptions that are objective and representative of the image content. Subjects were shown an image and a text box, and were asked to "Describe what is going on in the image". I asked subjects to capture the main aspects of the scene and provide descriptions that others are also likely to provide. This includes writing descriptions rather than "dialogs" or overly descriptive sentences. Workers were told that a good description should help others recognize the image from a collection of similar images. Instructions also mentioned that work with poor grammar would be rejected. Snapshots of the interface can be found in the appendix. Overall, I had 465 subjects for ABSTRACT-50S and 683 subjects for PASCAL-50S datasets. I ensured that each sentence for an image is written by a different subject. The average sentence length for the ABSTRACT-50S dataset is 10.59 words compared to 8.8 words for PASCAL-50S.

4.1.4 Experimental Setup

The goals of the experiments are two-fold:

- Evaluating how well the proposed metric CIDEr captures human judgement of consensus, as compared to existing metrics.
- Comparing existing state-of-the-art automatic image description approaches in terms of how well the descriptions they produce match human consensus of image descriptions.

I first describe how I selected candidate sentences for evaluation and the metrics I use for comparison to CIDEr. Then, I list the various automatic image description approaches I compare and the experimental set up.

Candidate Sentences: On ABSTRACT-50S, I use 48 of 50 sentences as reference sentences (sentence A in my triplet annotation). The remaining 2 sentences per image can be used as candidate sentences. I form 400 pairs of candidate sentences (B and C in the triplet annotation). These include two kinds of pairs. The first are 200 human–human correct pairs (HC), where I pick two human sentences describing the same image. The second kind are

200 human–human incorrect pairs (HI), where one of the sentences is a human description for the image and the other is also a human sentence but describing some other image from the dataset picked at random.

For PASCAL-50S, my candidate sentences come from a diverse set of sources: human sentences from the UIUC Pascal Sentence Dataset as well as machine-generated sentences from five automatic image description methods. These span both retrieval-based and generation-based methods: Midge (Mitchell, Han, and Hayes 2012), Babytalk (Kulkarni et al. 2011), Story (Farhadi et al. 2010), and two versions of Translating Video Content to Natural Language Descriptions (Rohrbach et al. 2013) (Video and Video+).[3] I form 4,000 pairs of candidate sentences (again, B and C for my triplet annotation). These include four types of pairs (1,000 each). The first two are human–human correct (HC) and human–human incorrect (HI) similar to ABSTRACT-50S. The third are human–machine (HM) pairs formed by pairing a human sentence describing an image with a machine generated sentence describing the same image. Finally, the fourth are machine–machine (MM) pairs, where I compare two machine generated sentences describing the same image. I pick the machine generated sentences randomly, so that each method participates in roughly equal number of pairs, on a diverse set of images.

For consistency, I drop two reference sentences for the PASCAL-50S evaluations so that I evaluate on both datasets (ABSTRACT-50S and PASCAL-50S) with a maximum of 48 reference sentences.

Metrics: The existing metrics used in the community for evaluation of image description approaches are BLEU (Papineni et al. 2002), ROUGE (Lin 2004) and METEOR (Lavie and Denkowski 2009). BLEU is precision-based and ROUGE is recall-based. More specifically, image description methods have used versions of BLEU called BLEU_1 and BLEU_4, and a version of ROUGE called ROUGE_1. A recent survey paper (Elliott and Keller 2014) has used a different version of ROUGE called ROUGE_S, as well as the machine translation metric

[3] I thank the authors of these approaches for making their outputs available to us.

called METEOR (Lavie and Denkowski 2009). I now briefly describe these metrics. Very recently, the SPICE metric was proposed by Anderson et al. 2016. I describe comparisons of CIDEr to this metric in the related work chapter (Chapter. 3).

BLEU (BiLingual Evaluation Understudy) (Papineni et al. 2002) is a popular metric for Machine Translation (MT) evaluation. It computes an n-gram based precision for the candidate sentence with respect to the references. The key idea of BLEU is to compute precision by *clipping*. Clipping computes precision for a word, based on the maximum number of times it occurs in any reference sentence. Thus, a candidate sentence saying "The The The", would get credit for saying only one "The", if the word occurs at most once across individual references. BLEU computes the geometric mean of the n-gram precisions and adds a brevity-penalty to discourage overly short sentences. The most common formulation of BLEU is BLEU4, which uses 1-grams up to 4-grams, though lower-order variations such as BLEU1 (unigram BLEU) and BLEU2 (unigram and bigram BLEU) are also used. Similar to (Elliott and Keller 2014; Hodosh, Young, and Hockenmaier 2013) for evaluating image descriptions, I compute BLEU at the sentence level. For machine translation BLEU is most often computed at the corpus level where correlation with human judgment is high; the correlation is poor at the level of individual sentences. In this paper we are specifically interested in computing the score for a metric given a single sentence (so that we can measure agreement with human consensus, which is available on pairs of sentences (see Sec. 4.1.4)).

ROUGE stands for Recall Oriented Understudy of Gisting Evaluation (Lin 2004). It computes n-gram based recall for the candidate sentence with respect to the references. It is a popular metric for summarization evaluation. Similar to BLEU, versions of ROUGE can be computed by varying the n-gram count. Two other versions of ROUGE are ROUGE$_S$ and ROUGE$_L$. These compute an F-measure with a recall bias using *skip-bigrams* and *longest common subsequence* respectively, between the candidate and each reference sentence. Skip-bigrams are all pairs of ordered words in a sentence, sampled non-consecutively. Given these scores, they return the maximum score across the set of references as the

judgment of quality. **METEOR** stands for Metric for Evaluation of Translation with Explicit ORdering (Lavie and Denkowski 2009). Similar to ROUGE$_L$ and ROUGE$_S$, it also computes the F-measure based on matches, and returns the maximum score over a set of references as its judgment of quality. However, it resolves word-level correspondences in a more sophisticated manner, using exact matches, stemming and semantic similarity. It optimizes over matches minimizing *chunkiness*. Minimizing chunkiness implies that matches should be consecutive, wherever possible. It also sets parameters favoring recall over precision in its F-measure computation. I implement all the metrics, except for METEOR, for which I use (Denkowski and Lavie 2014) (version 1.5). Similar to BLEU, I also aggregate METEOR scores at the sentence level.

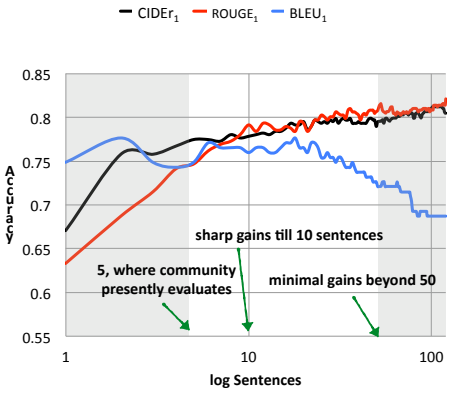
Machine Approaches: I comprehensively evaluate which machine generation methods are best at matching consensus sentences. For this experiment, I select a subset of 100 images from the UIUC Pascal Sentence Dataset for which I have outputs for all the five machine description methods used for evaluation: Midge (Mitchell, Han, and Hayes 2012), Babytalk (Kulkarni et al. 2011), Story (Farhadi et al. 2010), and two versions of Translating Video Content to Natural Language Descriptions (Rohrbach et al. 2013) (Video and Video+). For each image, I form all ${}^5C_2$ pairs of machine–machine sentences. This ensures that each machine approach gets compared to all other machine approaches on each image. This results in 1,000 pairs. I form triplets by "tripling" each pair with 20 random reference sentences. I collect human judgement of consensus using the triplet annotation modality as well as evaluate the proposed automatic consensus metric CIDEr using the same reference sentences. In both cases, I count the fraction of times a machine description method beats another method in terms of being more similar to the reference sentences. To the best of my knowledge, this is the first work to perform an exhaustive evaluation of automated image captioning, across retrieval- and generation-based methods.

4.1.5 Results

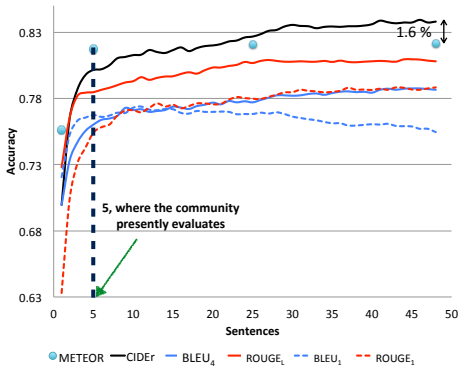
In this section I evaluate the effectiveness of our consensus-based metric CIDEr on the PASCAL-50S and ABSTRACT-50S datasets. I begin by exploring how many sentences are sufficient for reliably evaluating our consensus metric. Next, I compare our metric against several other commonly used metrics on the task of matching human consensus. Then, using CIDEr I evaluate several existing automatic image description approaches. Finally, I compare performance of humans and CIDEr at predicting consensus.

How many sentences are enough?

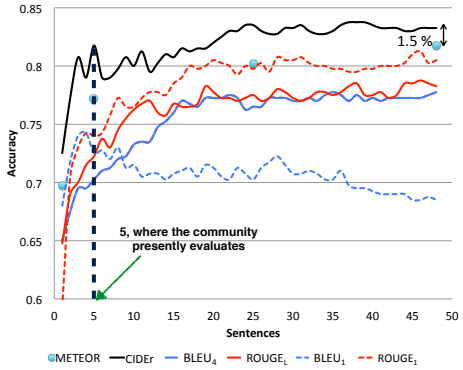
I begin by analyzing how the number of reference sentences affects the accuracy of automated metrics. To quantify this, I collected 120 sentences for a subset of 50 randomly sampled images from the UIUC Pascal Sentence Dataset. I then pooled human–human correct, human–machine, machine–machine and human–human incorrect sentence pairs (179 in total) and got triplet annotations. This gives us the ground truth consensus score for all pairs. I evaluate BLEU_1, ROUGE_1 and CIDEr_1 with up to 100 reference sentences used to score the candidate sentences. I find that the accuracy improves for the first 10 sentences (Fig. 4.2a) for all metrics. From 1 to 5 sentences, the agreement for ROUGE_1 improves from 0.63 to 0.77. Both ROUGE_1 and CIDEr_1 continue to improve until reaching 50 sentences, after which the results begin to saturate somewhat. Curiously, BLEU_1 shows a decrease in performance with more sentences. BLEU does a max operation over sentence level matches, and thus as more sentences are used, the likelihood of matching a lower quality reference sentence increases. Based on this pilot, I collected 50 sentences per image for the ABSTRACT-50S and PASCAL-50S datasets. For the remaining experiments I report results using 1 to 50 sentences.



(a) Pilot Performance



(b) PASCAL-50S



(c) ABSTRACT-50S

Figure 4.2: (**a**): I show accuracy (y-axis) versus *log* number of sentences (x-axis) for the pilot study. Note that the gains saturate after 50 sentences. **(b) and (c):** Accuracy of automated metrics (y-axis) plotted against number of reference sentences (x-axis) for PASCAL-50S (b) and ABSTRACT-50S (c). Metrics currently used for evaluating image descriptions are shown in *dashed* lines. Other existing metrics and our proposed metric are in **solid** lines. CIDEr is the best performing metric on both datasets followed by METEOR. METEOR is sampled at fewer points, due to high run-time. Note that more reference sentences that were collected clearly help.

Accuracy of Automated Metrics

I evaluate the performance of CIDEr, BLEU, ROUGE and METEOR at matching the human consensus scores in Fig. 4.2. That is, for each metric I compute the scores for two candidate sentences. The metric is correct if the sentence with higher score is the same as the sentence chosen by our human studies as being more similar to the reference sentences. The candidate sentences are both human and machine generated. For BLEU and ROUGE I show both their popular versions and the version we found to give best performance. I sampled METEOR at fewer points due to high run-time. For a more comprehensive evaluation across different versions of each metric, please see the appendix.

At 48 sentences, I find that CIDEr is the best performing metric, on both ABSTRACT-50S as well as PASCAL-50S. It is followed by METEOR on each dataset. Even using only 5 sentences, both CIDEr and METEOR perform well in comparison to BLEU and ROUGE. CIDEr beats METEOR at 5 sentences on ABSTRACT-50S, whereas METEOR does better at five sentences on PASCAL-50S. This is because METEOR incorporates soft-similarity, which helps when using fewer sentences. However, METEOR, despite its sophistication does a max across reference scores, which limits its ability to utilize larger numbers of reference sentences. Popular metrics like ROUGE_1 and BLEU_1 are not as good at capturing consensus. CIDEr provides consistent performance across both the datasets, giving 84% and 84% accuracy on PASCAL-50S and ABSTRACT-50S respectively.

Considering previous papers only used 5 reference sentences per image for evaluation, the relative boost in performance is substantial. Using BLEU_1 or ROUGE_1 at 5 sentences, we can obtained 76% and 74% accuracy on PASCAL-50S. With CIDEr at 48 sentences, we can achieve 84% accuracy. This brings automated evaluation much closer to human performance (90%, details in Sec. 4.1.5). On the Flickr8K dataset (Hodosh, Young, and Hockenmaier 2013) with human judgments on 1-5 ratings, METEOR has a correlation (Spearman's ρ) of 0.56 (Elliott and Keller 2014), whereas CIDEr achieves a correlation of

Table 4.1: Results on four kinds of pairs for PASCAL-50S and two kinds of pairs for ABSTRACT-50S. The best performing method is shown in **bold**. Note: I use ROUGE_L for PASCAL-50S and ROUGE_1 for ABSTRACT-50S

Metric	PASCAL-50S				ABSTRACT-50S	
	HC	HI	HM	MM	HC	HI
BLEU_4	64.8	97.7	93.8	63.6	65.5	93.0
ROUGE	66.3	98.5	95.8	64.4	**71.5**	91.0
METEOR	65.2	99.3	**96.4**	67.7	69.5	94.0
CIDEr	**71.8**	**99.7**	92.1	**72.2**	**71.5**	**96.0**

0.58 with human judgments.[4]

I next show the best performing versions of the metrics CIDEr, BLEU, ROUGE and METEOR on PASCAL-50S and ABSTRACT-50S, respectively, for different kinds of candidate pairs (Table 4.1). As discussed in Sec. 4.1.3 I have four kinds of pairs: (human–human correct) HC, (human–human incorrect) HI, (human–machine) HM, and (machine–machine) MM. I found that out of six cases, the proposed automated metric is best in five. The metric shows significant gains on the challenging MM and HC tasks that involve differentiating between fine-grained differences between sentences (two machine generated sentences and two human generated sentences). This result is encouraging because it indicates that the CIDEr metric will perform well as image description methods continue to improve. On the easier tasks of judging consensus on HI and HM pairs, all methods perform well.

Which automatic image description approaches produce consensus descriptions?

I have shown that CIDEr and the new datasets containing 50 sentences per image provide a more accurate metric over previous approaches. I now use it to evaluate some existing automatic image description approaches. The methodology for conducting this experiment is described in Sec. 4.1.4. The results are shown in Fig. 4.3. I show the fraction of times

[4]I thank Desmond Elliot for the result.

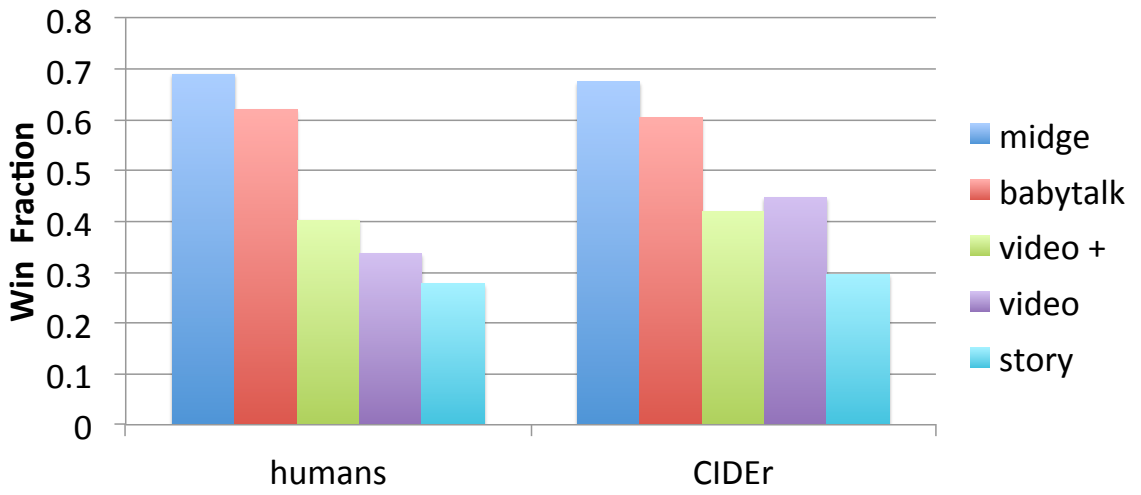


Figure 4.3: Fraction of times a machine generation approach wins against the other four (y-axis), plotted for human annotations and the proposed automated metric, CIDEr.

an approach is rated better than other approaches on the y-axis. Note that Midge (Mitchell, Han, and Hayes 2012) is rated as having the best consensus by both humans and CIDEr, followed by Babytalk (Kulkarni et al. 2011). Story (Farhadi et al. 2010) is the lowest ranked, by both humans and CIDEr. Humans and CIDEr differ on the ranking of the two video approaches (Video and Video+) (Rohrbach et al. 2013). I also calcuated the Pearson's correlation between the fraction of wins for a method on human annotations and using CIDEr. We find that humans and CIDEr agree with a high correlation (0.98).

Human Performance

In the final set of experiments I measure human performance at predicting which of two candidate sentences better matches the consensus. Human performance puts into context how clearly consensus is defined, and provides a loose bound on how well we can expect automated metrics to perform. I evaluate both human and machine performance at predicting consensus on all 4,000 pairs from PASCAL-50S dataset and 400 pairs from the ABSTRACT-50S dataset described in Sec. 4.1.4. To create the same experimental set up for both humans and machines, I obtained ground truth consensus for each of the pairs using the triplet annotation on 24 references out of 48. For predicting consensus, humans (via triplet annotations) and machines both use the remaining 24 sentences as reference sentences. I

find that the best machine performance is 82% on PASCAL-50S using CIDEr, in contrast to human performance which is at 90%. On the ABSTRACT-50S dataset, CIDEr is at 82% accuracy, whereas human performance is at 83%.

4.1.6 Gameability and Evaluation Server

Gameability When optimizing an algorithm for a specific metric undesirable results may be achieved. The "gaming" of a metric may result in sentences with high scores, yet produce poor results when judged by a human. To help defend against the future gaming of the CIDEr metric, I propose several modifications to the basic CIDEr metric called CIDEr-D.

First, I propose the removal of stemming. When performing stemming the singular and plural forms of nouns and different tenses of verbs are mapped to the same token. The removal of stemming ensures the correct forms of words are used. Second, in some cases the basic CIDEr metric produces higher scores when words of higher confidence are repeated over long sentences. To reduce this effect, I introduce a Gaussian penalty based on the difference between candidate and reference sentence lengths. Finally, the sentence length penalty may be gamed by repeating confident words or phrases until the desired sentence length is achieved. I combat this by adding clipping to the n-gram counts in the CIDEr$_n$ numerator. That is, for a specific n-gram I clip the number of candidate occurrences to the number of reference occurrences. This penalizes the repetition of specific n-grams beyond the number of times they occur in the reference sentence. These changes result in the following equation (analogous to Equation 4.2):

$$\text{CIDEr-D}_n(\mathbf{c}_i, S_i) = \frac{10}{m} \sum_j e^{\frac{-(l(\mathbf{c}_i)-l(\mathbf{s}_{ij}))^2}{2\sigma^2}} * \frac{\min(\boldsymbol{g}^{\boldsymbol{n}}(\mathbf{c}_i), \boldsymbol{g}^{\boldsymbol{n}}(\mathbf{s}_{ij})) \cdot \boldsymbol{g}^{\boldsymbol{n}}(\mathbf{s}_{ij})}{\|\boldsymbol{g}^{\boldsymbol{n}}(\mathbf{c}_i)\| \|\boldsymbol{g}^{\boldsymbol{n}}(\mathbf{s}_{ij})\|}, \tag{4.4}$$

Where $l(\mathbf{c}_i)$ and $l(\mathbf{s}_{ij})$ denote the lengths of candidate and reference sentences respectively. I use $\sigma = 6$. A factor of 10 is added to make the CIDEr-D scores numerically similar

to other metrics.

The final CIDEr-D metric is computed in a similar manner to CIDEr (analogous to Equation 4.3):

$$\text{CIDEr-D}(\mathbf{c}_i, S_i) = \sum_{n=1}^{N} w_n \text{CIDEr-D}_n(\mathbf{c}_i, S_i), \tag{4.5}$$

Similar to CIDEr, uniform weights are used. I found that this version of the metric has a rank correlation (Spearman's ρ) of 0.94 with the original CIDEr metric while being more robust to gaming. Qualitative examples of ranking can be found in the appendix.

Evaluation Server To enable systematic evaluation and benchmarking of image description approaches based on consensus, I have made CIDEr-D available as a metric in the COCO caption evaluation server (Chen et al. 2015).

4.2 Context-aware Captions from Context-agnostic Supervision

In this section, I will explore how to generate image captions that are sensitive to the context in which we wish to describe images. We will study two such real-world vision tasks that require pragmatic (contextual) reasoning. The first is *justification*, where the model needs to justify why an image corresponds to one fine-grained object category, as opposed to a closely related, yet undepicted category. Justification is a task that is important for hobbyists, and domain experts: ornithologists and botanists often need to explain why an image depicts particular species as opposed to a closely-related species. Another potential application for justification is in machine teaching, where an algorithm instructs non-expert humans about new concepts.

The second task is *discriminative image captioning*, where the goal is to generate a sentence that describes an image in context of other semantically similar images. This task is not only grounded in pragmatics, but is also interesting as a scene understanding task to check fine-grained image understanding. It also has potential applications to human robot interaction.

Recent work by Andreas and Klein 2016 derives pragmatic behaviour in neural language models using only context-free data. While I am motivated by similar considerations, the key algorithmic novelty of work presented in this section over Andreas and Klein 2016 is a unified inference procedure which leads to more efficient search for discriminative sentences (Sec. 4.2.3). The approach is based on the realization that one may simply re-use the sampling distribution from the generative model, instead of training a separate model to assess discriminativeness (Andreas and Klein 2016). This also has important implications for practitioners, since one can easily adapt existing context-free captioning models for context-aware captioning without additional training. Furthermore, while Andreas and Klein 2016 was applied to an abstract scenes dataset (Zitnick and Parikh 2013), I apply the proposed model to two qualitatively different real-image datasets: the fine-grained birds dataset CUB-200-2011 (Wah et al. 2011), and the COCO (Lin et al. 2014) dataset which contains real-life scenes with common objects. My evaluations on CUB-Justify, and human evaluation on COCO show that the proposed approach outperforms baseline approaches at inducing discrimination.

4.2.1 Approach

I describe my approach for inducing context-aware language for: 1) *justification*, where the context is another class, and 2) *discriminative image captioning*, where the context is a semantically similar image. For clarity, I first describe the formulation for justification, and then discuss a modification for discriminative image captioning.

In the justification task (Fig. 1.2 top), I wish to produce a sentence s, comprised of a sequence of words $\{s_i\}$, based on a given image I of a target concept c_t in the context of a distractor concept c_d. The produced justification should capture aspects of the image that discriminate between the target, and the distractor concepts. Note that images of the distractor class are not provided to the algorithm.

I first train a generic context-agnostic image captioning model (from here on referred to

as speaker) using training data from Reed et al. 2016c who collected captions describing bird images on the CUB-200-2011 (Wah et al. 2011) dataset. I condition the model on c_t in addition to the image. That is, I model $p(s|I, c_t)$. This not only helps produce better sentences (providing the model access to more information), but is also the cornerstone of the approach for discrimination (Sec. 4.2.1). The language models are recurrent neural networks which represent the state-of-the-art for language modeling across a range of popular tasks like image captioning (Vinyals et al. 2015; Xu et al. 2015), machine translation (Sutskever, Vinyals, and Le 2014) *etc*.

Reasoning Speaker

To induce discrimination in the utterances from a language model, it is natural to consider using a generator, or speaker, which models $p(s|I, c_t)$ in conjunction with a listener function $f(s, c_t, c_d)$ that scores how discriminative an utterance s is. The task of a pragmatic reasoning speaker RS, then, is to select utterances which are good sentences as per the generative model p, and are discriminative per f:

$$RS(I, c_t, c_d) = \arg\max_s \lambda p(s|I, c_t) + (1-\lambda) f(s, c_t, c_d) \tag{4.6}$$

where $0 \leq \lambda \leq 1$ controls the tradeoff between linguistic adequacy of the sentence, and discriminativeness.

A similar reasoning speaker model forms the core of the approach of Andreas and Klein 2016, where p, and f are implemented using multi-layer perceptrons (MLPs). As noted in Andreas and Klein 2016, selecting utterances from such a reasoning speaker poses several challenges. First, exact inference in this model over the exponentially large space of sentences is intractable. Second, in general one would not expect the discriminator function f to factorize across words, making joint optimization of the reasoning speaker objective difficult. Thus, Andreas and Klein 2016 adopt a sampling based strategy, where p

is considered as the proposal distribution whose samples are ranked by a linear combination of p, and f (Eqn. 4.6). Importantly, this distribution is over full sentences, hence the effectiveness of this formulation depends heavily on the distribution captured by p, since the search over the space of all strings is solely based on the speaker. This is inefficient, especially when there is a mismatch in the statistics of the context-free (generative), and the unknown context-aware (discriminative) sentence distributions. In such cases, one must resort to drawing many samples to find good discriminative sentences.

Introspective Speaker

My approach for incorporating contextual behavior is based on a simple modification to the listener f (Eqn. 4.6). Given the generator p, I construct a listener module that wants to discriminate between c_t, and c_d, using the following log-likelihood ratio:

$$f(s, c_t, c_d) = \log \frac{p(s|c_t, I)}{p(s|c_d, I)}. \tag{4.7}$$

This listener only depends on a generative model, $p(s|c, I)$, for the two classes c_t, and c_d. I name it "introspector" to emphasize that this step re-uses the generative model, and does not need to train an explicit listener model. Substituting the introspector into Eqn. 4.6 induces the following introspective speaker model for discrimination:

$$\underbrace{\Delta(I, c_t, c_d)}_{\text{introspective speaker}} = \arg\max_s \lambda \underbrace{\log p(s|c_t, I)}_{\text{speaker}} + (1-\lambda) \underbrace{\log \frac{p(s|c_t, I)}{p(s|c_d, I)}}_{\text{introspector}} \quad , \tag{4.8}$$

with λ that trades-off the weight given to generation, and introspection (similar to Eqn. 4.6). In general, one can expect this approach to provide sensible results when c_t, and c_d are similar. That is, we expect humans to describe similar concepts in similar ways, hence $p(s|c_t, I)$ should not be too different from $p(s|c_d, I)$. Thus, the introspector is less likely to overpower the speaker in Eqn. 4.8 in such cases (for a given λ). Note that for sufficiently

different concepts the speaker alone is likely to be sufficient for discrimination. That is, describing the concept in isolation is likely to be enough to discriminate against a different or unrelated concept.

A careful inspection of the introspective speaker model reveals two desirable properties over previous work (Andreas and Klein 2016). First, the introspector model does not need training, since it only depends on p, the original generative model. Thus, existing language models can be readily re-used to produce context-aware outputs by conditioning on c_d. I demonstrate empirical validation of this in Sec. 4.2.3. This would help scale this approach to scenarios where it is not known apriori which concepts need to be discriminated, in contrast to approaches which train a separate listener module. Second, it leads to a unified, and efficient inference for the introspective speaker (Eqn. 4.8), which I describe next.

Emitter-Suppressor (ES) Beam Search for RNNs

I now describe a search algorithm for implementing the maximization in Eqn. 4.8, which I call *emitter-suppressor* (ES) beam search. I use the beam search (Lee, Hon, and Reddy 1990) algorithm, which is a heuristic graph-search algorithm commonly used for inference in Recurrent Neural Networks (Vijayakumar et al. 2018). See Chapter. 2 for a refresher on beam search.

I first factorize the posterior log-probability terms in the introspective speaker equation (Eqn. 4.8) $p(s|c_t, I) = \prod_{\tau=1}^{T} p(s_\tau|s_{1:\tau-1}, c_t, I)$, denoting $s_{1:T} = \{s_\tau\}_{\tau=1}^{T}$ ($s_{1:0}$ corresponds to a null string). T is the length of the sentence. I then combine terms from Eqn. 4.8, yielding the following emitter-suppressor objective for the introspective speaker:

$$\Delta(I, c_t, c_d) = \arg\max_{s} \sum_{\tau=1}^{T} \log \frac{\overbrace{p(s_\tau|s_{1:\tau-1}, c_t, I)}^{\text{emitter}}}{\underbrace{p(s_\tau|s_{1:\tau-1}, c_d, I)^{1-\lambda}}_{\text{suppressor}}}. \tag{4.9}$$

The emitter (numerator in Eqn. 4.9) is the generative model conditioned on the target concept

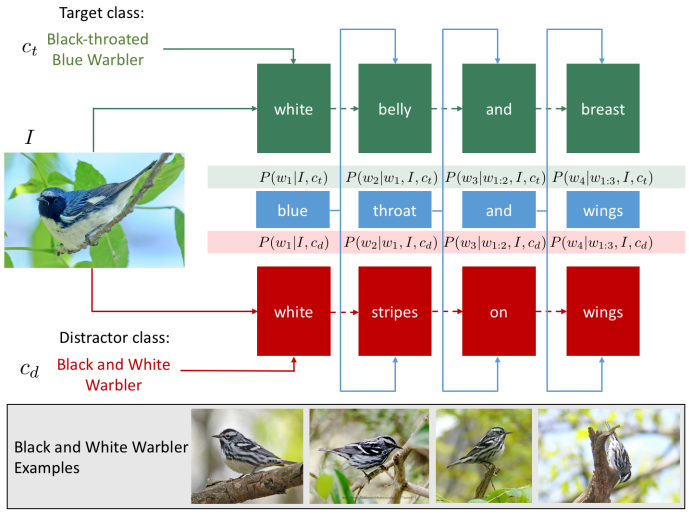


Figure 4.4: Emitter-suppressor beam search for beam size 1, for distinguishing an image of "black-throated blue warbler" from the distractor class "black and white warbler". Green: A language model $p(s|c_t, I)$ produces a caption "white belly and breast ... ". Red: When feeding the distractor class to the language model, since the two birds share the attribute white belly, which appears in the image, the term "white" is highly suppressed. Blue: Picking likely words for the emitter, and unlikely for the suppressor yields a discriminative caption "blue throat ..". Note that emitter, and suppressor share history (the previouly generated words).

c_t, deciding which token to select at a given timestep. The suppressor (the denominator in Eqn. 4.9) is conditioned on the distractor concept c_d, providing signals to the emitter on which tokens to avoid. This is intuitive – to be discriminative, we want to emit words that match c_t, but avoid emitting words that match c_d.

I maximize the emitter-suppressor objective (Eqn. 4.9) using beam search. Vanilla beam search, as typically used in language models, prunes the output space at every time-step keeping the top-B (usually incomplete) sentences with highest log-probabilities so far (speaker in Eqn. 4.8). Instead, I run beam search to keep the top-B sentences with highest ES ratio in Eqn. 4.9. Fig. 4.4 illustrates this ES beam search for a beam size of 1.

It is important to consider how the trade-off parameter λ affects the produced sentences. For $\lambda = 1$, the model generates descriptions that ignore the context. At the other extreme, low λ values are likely to make the produced sentences very different from any sentence in the training set (repeated words, ungrammatical sentences). It is not trivial to assume

that there exists a wide enough range of λ creating sentences that are both discriminative, and well-formed. However, our results (Sec. 4.2.3) indicate that such a range of λ exists in practice.

Discriminative Image Captioning

We are given a target image I_t, and a distractor I_d,that we wish to distinguish, similar to the two classes for the justification task. We will construct a speaker (or generator) for this task by training a standard image captioning model. Given this speaker, we can construct an emitter-suppressor equation (as in Eqn. 4.9):

$$\Delta(I_t, I_d) = \arg\max_{s} \sum_{\tau=1}^{T} \log \frac{\overbrace{p(s_\tau | s_{1:\tau-1}, I_t)}^{\text{emitter}}}{\underbrace{p(s_\tau | s_{1:\tau-1}, I_d)^{1-\lambda}}_{\text{suppressor}}}. \tag{4.10}$$

I re-use the mechanics of emitter-suppressor beam search from Sec. 4.2.1, conditioning the emitter on the target image I_t, and the suppressor on the distractor image I_d.

4.2.2 Experimental Setup

I provide details of the CUB dataset, of our CUB-Justify dataset used for evaluation, and of the speaker-training setup for the justification task. I then discuss the experimental protocols for discriminative image captioning.

Justification

CUB Dataset: The Caltech UCSD birds (CUB) dataset (Wah et al. 2011) contains 11788 images for 200 species of North American birds. Each image in the dataset has been annotated with 5 fine-grained captions by Reed *et.al* (Reed et al. 2016c). These captions mention various details about the bird ("This is a white spotted bird with a long pointed black beak.") while not mentioning the name of the bird species.

CUB-Justify Dataset: I collected a new dataset (CUB-Justify) with ground truth justifications for evaluating justification. I first sampled the target, and distractor classes from within a hyper-category created based on the last name of the folk names of the 200 species in CUB. For instance, "rufous hummingbird", and "ruby throated hummingbird" both fall in the hyper-category "hummingbird". I induced 37 such hyper-categories. The largest single hypercategory is "Warbler" with 25 categories. I then selected a subset of (approx.) 15 images from the test set of CUB-200-2011 (Wah et al. 2011) for each of the 200 classes, to form a CUB-Justify test split. I use the rest for speaker training (CUB-Justify train split).

Workers were then shown an image of the "rufous hummingbird", for instance, and a set of 6 other images (from CUB-Justify test split) all belonging to the distractor class "ruby throated hummingbird", to form the visual notion of the distractor class. They were also shown a diagram of the morphology of birds indicating various parts such as tarsus, rump, wingbars *etc* (similar to Reed et al. 2016c). The instruction was to describe the target image such that it is not confused with images from the distractor class. Some birds are best distinguished by non-visual cues such as their call, or their migration patterns. Thus, I drop the categories of birds from the original list of triplets which were labeled as too hard to distinguish by the workers. At the end of this process I was left with 3161 triplets with 5 captions each. I split this dataset into 1070 validation (for selecting the best value of λ), and 2091 test examples respectively. More details on the interface can be found in the appendix.

Speaker Training: I implement a model similar to "Show, Attend, and Tell" from Xu *et.al* (Xu et al. 2015), modifying the original model to provide the class as input, similar in spirit to (Hendricks et al. 2016a). Exact details of our model architecture are given in the appendix. I train the model on the CUB-Justify train split. Recall that this just has context-agnostic captions from (Reed et al. 2016c).

To evaluate the quality of our speaker model, I report numbers here using the CIDEr-D metric (Vedantam, Lawrence Zitnick, and Parikh 2015) commonly used for image captioning (Hendricks et al. 2016a; Karpathy and Fei-Fei 2015; Vinyals et al. 2015) computed on

the context-agnostic captions from (Reed et al. 2016c). My captioning model with both the image, and class as input reaches a validation score of 50.2 CIDEr-D, while the original image-only captioning model reaches a CIDEr-D of 49.1. The scores are in a similar range as existing CUB captioning approaches (Hendricks et al. 2016a).

Justification Evaluation: I measure performance of the (context-aware) justification captions on the CUB-Justify discriminative captions using the CIDEr-D metric. CIDEr-D weighs n-grams by their inverse document frequencies (IDF), giving higher weights to sentences having "content" n-grams ("red beak") than generic n-grams ("this bird") (Hendricks et al. 2016a). Further, CIDEr-D captures importance of an n-gram for the image. For instance, it emphasizes "red beak" over, say, "black belly" if "red beak" is used more often in human justifications. I also report METEOR (Lavie and Denkowski 2009) scores for completeness. More detailed discussion on metrics can be found in the appendix.

Discriminative Image Captioning

Dataset: I want to test if reasoning about context with an introspective speaker can help discriminate between pairs of very similar images from the COCO dataset. To construct a set of confusing image pairs, I follow two strategies. First, *easy confusion*: For each image in the validation (test) set, I find its nearest neighbor in the last fully connected feature map of a pre-trained VGG-16 CNN (Simonyan and Zisserman 2015), and repeat this process of neighbor finding for 1000 randomly chosen source images. Second, *hard confusion*: To further narrow down to a list of semantically similar confusing images, I then run the speaker model on the nearest neighbor images, and compute word-level overlap (intersection over union) of their generated sentences. I then pick the top 1000 pairs with most overlap. Interestingly, the top 539 pairs had identical captions. This reflects the issue of the output of image captioning models lacking diversity, and seeming templated (Vinyals et al. 2015).

Speaker Training and Evaluation: I train the generative speaker for use in emitter-suppressor beam search using the model from (Vinyals et al. 2015) implemented in the

neuraltalk2 project (*Neuraltalk2 Image Captioning*). I use the train/val/test splits from (*Neuraltalk2 Image Captioning*). My trained and finetuned speaker model achieves a performance of 91 CIDEr-D on the test set. As seen in Eqn. 4.10, no category information is used for this task. I evaluate approaches for discriminative image captioning based on how often they help humans to select the correct image out of the pair of images.

4.2.3 Results

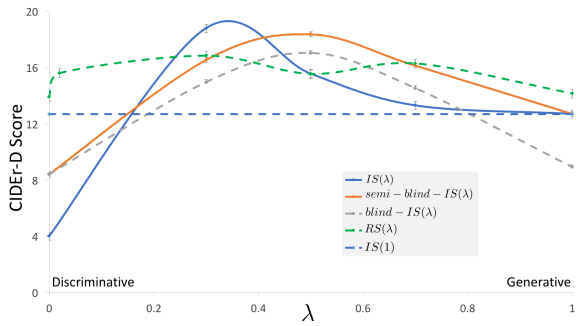


Figure 4.5: **CUB-Justify validation results:** CIDEr-D vs. λ on CUB-Justify validation. The introspective speaker approaches (IS(λ) and semi-blind-IS(λ)) models perform best, followed by the class-only introspective speaker (blind-IS(λ)). semi-blind-IS(λ) outperforms other methods for a wider range of λ. All approaches which reason about pragmatics beat the baseline generative approach IS(1). Error bars denote standard error of the mean score estimated across the validation set.

Justification

Methods and Baselines: I evaluate the following models: **1.** IS(λ): Introspective speaker from Eqn. 4.8; **2.** IS(1): standard literal speaker, which generates a caption conditioned on the image and target class, but which ignores the distractor class; **3.** semi-blind-IS(λ): Introspective speaker in which the listener does not have access to the image, but the speaker does; **4.** blind-IS(λ): Introspective speaker without access to image, conditioned only on classes; **5.** RS(λ): My implementation of Andreas and Klein 2016, but using our (more powerful) language model, and Eqn. 4.8 with a listener that models $\frac{p(s|c_t)}{p(s|c_d)}$ (similar to semi-blind-IS(λ)) for ranking samples (as opposed to a trained MLP (Andreas and Klein 2016),

to keep things comparable). All approaches use 10 beams/samples (which is better than lower values) unless stated otherwise.

Validation Performance: Fig. 4.5 shows the performance on CUB-Justify validation set as a function of λ, the hyperparameter controlling the tradeoff between the speaker and the introspector (Eqn. 4.8). For the RS(λ) baseline, λ stands for the tradeoff between the log-probability of the sentence and the score from the discriminator function for sample re-ranking. A few interesting observations emerge. First, both our IS(λ) and semi-blind-IS(λ) models outperform the baselines for the mid range of λ values. IS(λ) model does better overall, but semi-blind-IS(λ) has a more stable performance over a wider range of λ. This indicates that when conditioned on the image, the introspector has to be highly discriminative (low lambda values) to overcome the signals from the image, since discrimination is between classes.

Second, as λ is decreased from 1, most methods improve as the sentences become more discriminative, but then get worse again as λ becomes too low. This is likely to happen because when λ is too low, the model explores rare tokens and parts of the output space that have not been seen during training, leading to badly-formed sentences (Fig. 4.6). This effect is stronger for IS(λ) models than for RS(λ), since RS(λ) searches the output space over samples from the generator and only ranks using the joint reasoning speaker objective (Eqn. 4.6). Interestingly, at $\lambda = 1$ (no discrimination), the RS(λ) approach, which samples from the generator, also performs better than other approaches, which use beam search to select high log-probability (context-agnostic) sentences. This indicates that in the absence of ground truth justifications, there is indeed a discrepancy between searching for discriminativeness and searching for a highly likely context-agnostic sentence.

I performed more comparisons with the RS(λ) baseline, sweeping over $\{10, 50, 100\}$ samples from the generator for listener reranking (Eqn. 4.6). I found that using 100 samples, RS(λ) gets comparable CIDEr-D scores (18.8) (but lower METEOR scores) than the semi-blind-IS(λ) approach with a beam size of 10. This suggests that the semi-blind-IS(λ)

Table 4.2: **CUB-Justify test results:** CIDEr-D, and METEOR scores (higher the better) computed on test set of CUB-Justify. Each model used the best λ selected on the validation set (Fig. 4.5). Error values are standard error of the mean (SEM is less than 0.05 for METEOR). semi-blind-IS(λ) outperforms other methods.

Approach	CIDEr-D	METEOR
IS(λ)	18.4 ± 0.2	26.5
semi-blind-IS(λ)	$\mathbf{18.5 \pm 0.2}$	**27.5**
RS(λ)	15.8 ± 0.2	26.5
IS(1)	12.3 ± 0.1	25.3
blind-IS(λ)	16.1 ± 0.2	26.8

approach is more computationally efficient at exploring the output space because the emitter-suppressor beam search allows my approach to do joint greedy inference over speaker and introspector, leading to more meaningful local decisions. For completeness, I also trained a listener module discriminatively, and used it as a ranker for RS(λ). I found that this gets to 16.2 ± 0.3 CIDEr-D (at $\lambda = 0.5$) on validation, which is lower than IS(λ), showing that the bottleneck for performance is sampling, rather than the discriminativeness of the listener. More details can be found in the appendix.

Test Performance: Table. 4.2 details the performance of the above models on the test set of CUB-Justify, with each model using its best-performing λ on the validation set (Fig. 4.5). Both introspective-speaker models strongly outperform the baselines, with semi-blind-IS(λ) slightly outperforming the IS(λ) model. This could be due to the performance of semi-blind-IS(λ) being less sensitive to the exact choice of λ (from Fig. 4.5). Among the baselines, the best performing method is the blind-IS(λ) model, presumably because this model does emitter-suppressor beam search, while the other two baseline approaches rely on sampling and regular beam search respectively.

Qualitative Results: I next showcase some qualitative results that demonstrate 1) aspects of pragmatics, and 2) context dependence captured by our best-performing semi-blind-IS(λ) model. Fig. 4.6 demonstrates how sentences uttered by the introspective speaker change with λ. At $\lambda = 1$ the sentence describes the image well, but is oblivious of the context (distractor

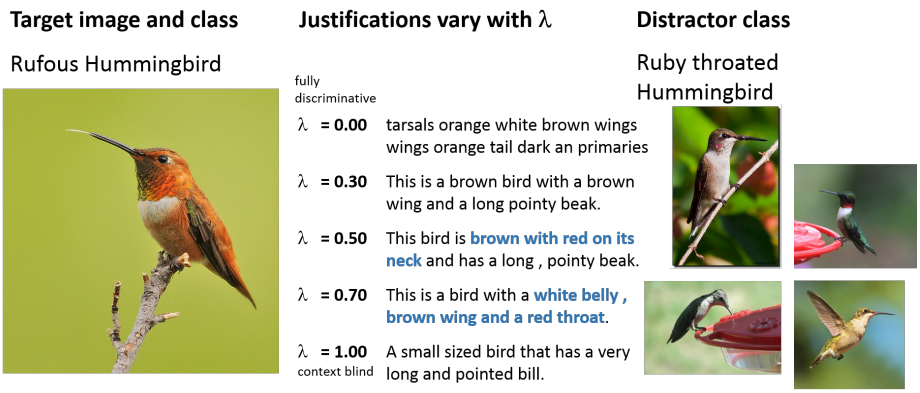


Figure 4.6: **The effect of context weight**: An image of a "Rufous Hummingbird" in the context of another hummingbird type. A generative (context-blind) description describes the bird as having a long beak, but this feature is not discriminative. When taking into account the context, intermediate λ values yield descriptions that highlight that the Rufous is brown with a red throat. For $\lambda = 0$, the model does not force sentences to be well formed.

class). The sentence "A small sized bird has a very long and pointed bill." is discriminative of hummingbirds against other birds, but not among hummingbirds (many of which tend to have long beaks/bills). At $\lambda = 0.7$, and $\lambda = 0.5$, the model captures discriminative features such as the "red neck", "white belly", and "red throat". Interestingly, at $\lambda = 0.7$ the model avoids saying "long beak", a feature shared by both birds. Next, Fig. 4.7 demonstrates how the selected utterances change based on the context. A limitation of our approach is that, since the model never sees discriminative training data, in some cases it produces repeated words ("green green green") when encouraged to be discriminative at inference time.

Finally, Fig. 4.8 illustrates the importance of visual reasoning for the justification task. Fine-grained species often have large intra-class variances which a *blind* approach to justification would ignore. Thus, a good justification approach needs to be grounded in the image signal to pick the discriminative cues appropriate for the given instance.

Discriminative Image Captioning

As explained in Sec. 4.2.2 I create two sets of semantically similar target, and distractor images: *easy confusion* based on fully connected layer (FC7) features alone, and *hard confusion* based on both FC7, and sentences generated from the speaker (image captioning

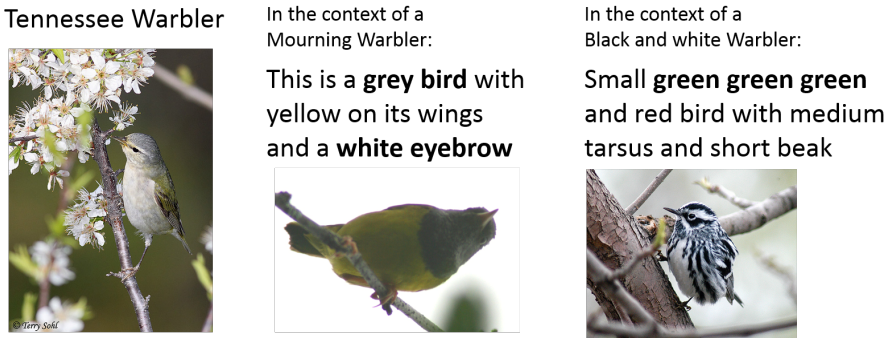


Figure 4.7: **The effect of context class**: An image of a "Tennessee Warbler", which has light green wings, and a white eyebrow. When described in the context of a mourning warbler, which has a green hue, the description highlights that the target bird has a white eyebrow. When described in the context of the "Black and White Warbler", the description highlights that the target bird has green color.

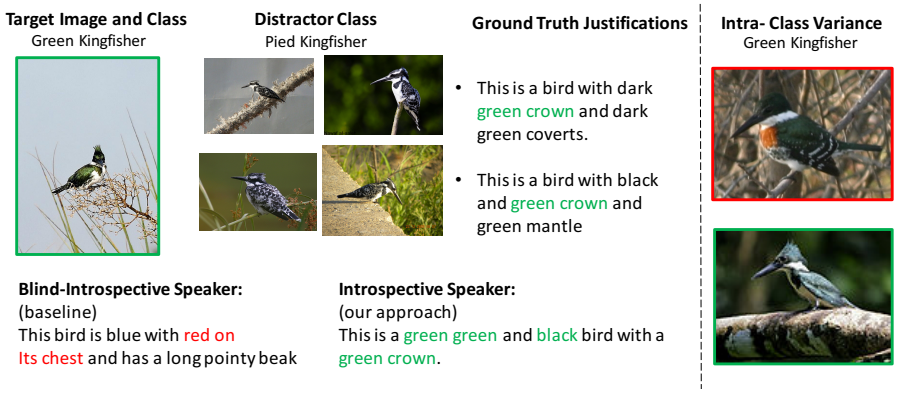


Figure 4.8: **The importance of visual signal for justification in fine-grained categories**. Given the image of a green kingfisher (left), a blind-IS(λ) model says the bird has "red on its chest", which is inaccurate for this image, and a "long pointy beak", which is not a discriminative feature for this context. At the same time, the semi-blind-IS(λ) model mentions the "green crown", and avoids uttering "red chest". Given the complicated intra-category invariances in bird categories (right), it is intuitive that the image signal is important for justification.

Figure 4.9: Pairs of images whose captions generated by a generic captioning speaker baseline (S) are identical. We apply our introspective speaker (IS) technique to distinguish the image on the left from the image on the right in each pair. The target image (left) is shown with a green border when the IS generated sentence is able to identify it correctly. Notice how the introspective speaker often refers more unambiguously to the target image. For example, for the sheep image (middle left), the IS generated sentence mentions that the sheep are grazing in a lush green field. In the bottom row I show some failure examples. The bottom left example is interesting, where the model calls the stop sign a policeman. In some cases (the wedding cake image), where the distributions captured by the emitter, and supressor RNN's are identical, our IS approach produces the same sentence as the baseline (S).

model). I am interested in understanding if emitter-suppressor inference helps identify the target image better than the generative speaker baseline. Thus the two approaches are speaker (S) (baseline), and introspective speaker (IS) (our approach). I use $\lambda = 0.3$ based on the results on the CUB dataset. I run all approaches at a beam size of 2 (typically best for COCO (*Neuraltalk2 Image Captioning*)).

Human Studies: I setup a two annotation forced choice (2AFC) study where I show a caption to raters asking them to "pick an image that the sentence is more likely to be describing.". Each target distractor image pair is tested against the generated captions. I check the fraction of times a method caused the target image to be picked by a human. A discriminative image captioning method is considered better if it enables humans to identify the target image more often. Results of the study are summarized in Table. 4.3. I found that our approach outperforms the baseline speaker (S) on the *easy confusion* as well as the *hard confusion* splits. However, the gains from our approach are larger on the *hard confusion* split, which is intuitive.

Qualitative Results: The qualitative results from the COCO experiments are shown in Fig. 4.9. The target image, when successfully identified, is shown with a green border. I show examples where the model identifies the target image better in the first two rows, and some failure cases in the third row. Notice how the model is able to modify its utterances to account for context, and pragmatics, when going from $\lambda = 1$ (speaker) to $\lambda = 0.3$ (introspective speaker). Note that the sentences typically respect grammatical constructs despite being forced to be discriminative.

4.2.4 Discussion

Describing absence of concepts and inducing comparative language are exciting directions for future work on justification. For instance, when justifying why an image is a lion and not a tiger, it would be useful to be able to say "because it does not have stripes.", or "because it has a more hair on its face." Beyond pragmatics, the justification task also has interesting

Table 4.3: % of image pairs that are correctly discriminated by humans, based on descriptions in COCO. Introspective speaker (IS) is better at pointing to the target image given a confusing distractor image across both easy, and hard data splits than a speaker (S). Standard error is below the precision I report numbers at.

Approach	*easy confusion* (%)	*hard confusion* (%)
S (baseline)	74.6	52.5
IS (ours)	**89.0**	**74.1**

relations to human learning. Indeed, we all experience that we learn better when someone takes time out to justify or explain their point of view. One can imagine such justifications being helpful for "machine teaching", where a teacher (machine) can provide justifications to a human learner explaining the rationale for an image belonging to a particular fine-grained category as opposed to a different, possibly mistaken, or confusing fine-grained category.

There are some fundamental limitations to inducing context-aware captions from context-agnostic supervision. For instance, if two distinct concepts are very similar, human-generated context-free descriptions may be identical, and the proposed model (as well as baselines) would fail to extract any discriminative signal. Indeed, it is hard to address such situations without context-aware ground truth.

I believe modeling higher-order reasoning (such as pragmatics) by reusing the sampling distribution from language models can also be a powerful tool. It may be applicable to other higher-order reasoning, without necessarily setting up policy gradient estimators on reward functions. Indeed, the inference objective in this chapter can also be formulated for training. However, initial experiments on this did not yeild significant performance improvements.

CHAPTER 5
GROUNDING

In this chapter, we will study grounding and how it can help commonsense reasoning. The rationale for why grounding is useful is based on the idea that symbols have no intrinsic meaning of their own, other than when they are "grounded" or associated with something concrete in our physical world. In particular, through the course of this chapter, we will focus on approaches which perform grounding into abstract scene images made of clipart (Fig. 1.3). We are use abstract scenes because we are interested in reasoning about fine-grained notions of grounding (such as understanding that "wanting" something can imply "looking-at" something), which is hard to do with real images despite recent progress in computer vision.

We will first study the premise that grounding could potentially help improve commonsense reasoning in the first section, and develop preliminary modeling techniques which will ground text into vision by learning a similarity function using aligned data. At inference we will use this visual similarity with textual similarity to predict how plausible certain assertions about the world are, and whether our predictions of plausibility match human annotations. In the second section, we will build on top of the first work and learn word embeddings grounded in abstract scenes. The benefit of learning embeddings turns out to be that they not only translate to better performance, but also help make inference more flexible by removing the need to have visual signal to compute visual similarity. That is, one can implicitly reason about visual grounding by simply computing similarity between *visually grounded* word embeddings.

5.1 Learning Common Sense Via. Visual Abstraction

Teaching machines common sense has been a longstanding challenge at the core of Artificial Intelligence (AI) (Davis, Shrobe, and Szolovits 1993). Consider the task of assessing how plausible it is for a dog to jump over a tree. One approach is to mine text sources to estimate how frequently the concept of dogs jumping over trees is mentioned. A long history of works address the problem is this manner by mining knowledge from the web (Carlson et al. 2010; Hoffart et al. 2013; Lehmann et al. 2010) or by having humans manually specify facts (Bollacker et al. 2008b; Miller 1992; Singh et al. 2002; Speer and Havasi 2012) in text. Unfortunately, text is known to suffer from a reporting bias – namely that we generally talk about interesting and noteworthy things in text. Unfortunately, a lot of commonsense knowledge is fairly mundane, which makes it hard to learn using text alone.

While unwritten, commonsense knowledge is not unseen! The visual world around us is full of structure modeled by our commonsense knowledge. By reasoning visually about a concept we may be able to estimate its plausibility more accurately.

Unfortunately, extracting commonsense knowledge from visual content requires automatic and accurate detection of objects, their attributes, poses, and interactions. These remain challenging problems in computer vision. My key insight is that commonsense knowledge may be gathered from a high-level semantic understanding of a visual scene, and that low-level pixel information is typically unnecessary. In other words, photorealism is not necessary to learn common sense. In this section, I explore the use of human-generated abstract scenes made from clipart for learning common sense. Note that abstract scenes are inherently *fully* annotated, allowing us to exploit the structure in the visual world, while bypassing the difficult intermediate problem of training visual detectors.

Specifically, I consider the task of assessing the plausibility of an interaction or relation between a pair of nouns, as represented by a tuple (primary noun, relation, secondary noun) e.g., (boy, kicks, ball). As training data, I collect a dataset of tuples and their abstract

visual illustrations made from clipart, while the test tuples are extracted using information extraction tools applied to sentences from the COCO dataset, which contains real images. Thus, I show that the knowledge one can learn from abstract scenes generalizes to real images.

The abstract scene illustrations are created by subjects on Amazon Mechanical Turk (AMT). I use this to learn a scoring function that can score how well an abstract visual illustration matches a test tuple.

Given a previously unseen tuple, I assess its plausibility using both visual and textual information. A tuple is deemed plausible if it has high alignment with the training tuples and visual abstractions. When measuring textual similarity between tuples I exploit the significant progress that has been made in learning word similarities from web scale data using neural network embeddings (Mikolov et al. 2013; Pennington, Socher, and Manning 2014). A tuple's alignment with the visual abstractions provides information on its visual plausibility. I model a large number of free form relations (213) and nouns (2466), which may form over $\approx$1 billion possible tuples. I show that by jointly reasoning about text and vision, one can assess the plausibility of commonsense assertions more accurately than by reasoning about text alone.

In the next subsection, I will discuss the dataset of tuples that I collected for modeling commonsense.

5.1.1 Datasets

Abstract Scenes Vocabulary

In order to learn comprehensive commonsense knowledge, it is important for the library of clipart pieces to be expressive enough to model a wide variety of scenarios. Previous works on using visual abstractions depicted a boy and a girl playing in a park (Fouhey and Zitnick 2014; Zitnick and Parikh 2013; Zitnick, Parikh, and Vanderwende 2013) with a library of 58 objects, or fine-grained interactions between two people (Antol, Zitnick, and Parikh 2014)

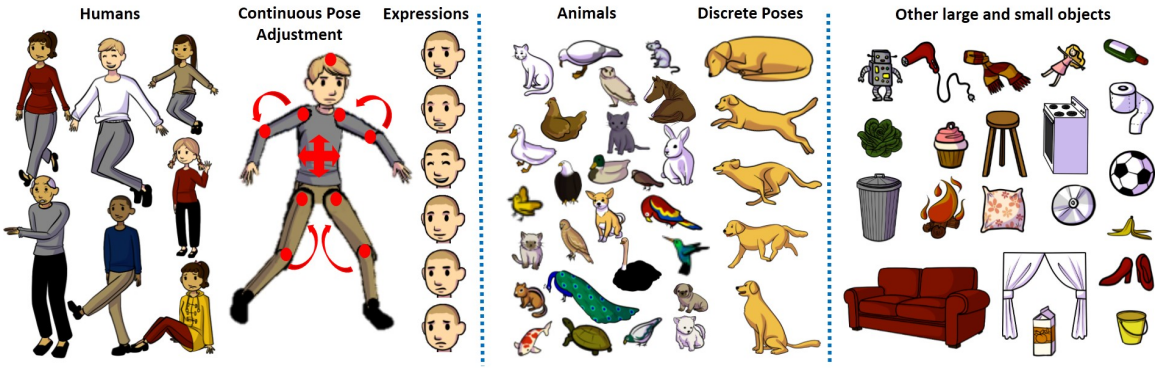


Figure 5.1: A subset of objects from our clipart library.

(no additional objects). Instead, our clipart library allows us to depict a variety of indoor scenes. It contains 20 "paperdoll" human models (Antol, Zitnick, and Parikh 2014) spanning genders, races, and ages with 8 different expressions. The limbs are adjustable to allow for continuous pose variations. The vocabulary contains over 100 small and large objects and 31 animals in various poses, that can be placed at one of 5 discrete scales or depths in the scene, facing left or right. Our clipart is also more realistic looking than previous work. A snapshot of the library can be viewed in Figure 5.1. Note that while this work restricts itself to indoor scenes, our idea is general and applicable to other scenes as well. More clipart objects and scenes can be easily added to the clipart library.

Tuple Extraction

Extracting Seed Assertions: To collect a dataset of commonsense assertions, I start by extracting a set of seed tuples from image captions. I use the COCO training set (Lin et al. 2014) containing images annotated with 80 object categories and five captions per image. I pick a subset of 9913 images whose annotated objects all come from a list of manually selected objects from our library of clipart.[1] Note that COCO images are not fully annotated and contain many more objects than those annotated. As a result, captions for these images could contain nouns that may not be part of the annotated object list or our clipart library.

[1]List: *person*, *cat*, *dog*, *frisbee*, *bottle*, *wine glass*, *cup*, *fork*, *knife*, *spoon*, *apple*, *sandwich*, *hotdog*, *pizza*, *cake*, *chair*, *couch*, *potted plant*, *bed*, *dining table*, *tv*, *book*, *scissors*, *teddy bear* was selected to capture objects in our clipart library that are commonly found in living room scenes.

Our model can handle this by using word embeddings as described in Sec. 5.1.2.

I split the images into VAL (4956 images) and TEST (4957 images). I then run the ReVerb (Etzioni et al. 2011) information extraction tool on the captions for these images (images are not involved anymore), along with some post-processing (described in the appendix) to obtain a set of (t_P, t_R, t_S) tuples, where t_P is the primary noun, t_R is the relation, and t_S is the secondary noun in the tuple t e.g., (plate, topped with, meat). All tuples containing relations that occur less than four times in the dataset are likely to be noisy extractions, and are removed. This gives us a set of 4848 tuples in VAL and 4778 in TEST, 213 unique relations in VAL and 204 in TEST, and 2466 unique nouns in VAL and 2378 in TEST. VAL and TEST have 893 tuples, 814 nouns, and 151 relations in common. These tuples form our seed commonsense assertions.

Expanding Seed Assertions: I expand the seed set of assertions by generating random assertions. This is done on both TEST and VAL independently. I iterate through each tuple twice, and pair the corresponding t_R with a random t_P and t_S from all nouns that occur at least 10 times[2]. So there are twice as many expanded tuples as there are seed tuples. This results in 9700 expanded tuples in VAL and 9554 in TEST. Note that I am sampling from a space of 160 primary nouns (>10 occurrences) $\times$ 204 relations $\times$ 160 nouns i.e., >5 *million* possible TEST assertions. In total across seed and expanded, the VAL set contains 14548 commonsense assertions spanning 213 relations, and the TEST set contains $14,332$ commonsense assertions spanning 204 relations. To the best of my knowledge, this is the first work that models such a large number of relations and commonsense assertions using vision.

Supervision on Expanded Assertions: I then showed the set of assertions (seed + expanded) to subjects on Amazon Mechanical Turk (AMT). I asked them to indicate if the scenario described by the assertion is typical or not. They were also given an option to flag scenarios that make no sense. I collected 10 judgments per assertion. A snapshot of this

[2]This is a coarse proxy for sampling nouns proportional to how often they occur in the seed set.

interface can be found in the appendix.

80.1% of annotations on seed tuples were positive. This is not surprising because these tuples were extracted from descriptions of images, and were thus clearly plausible. The creation of random expanded tuples predominantly adds negatives. But we found that some randomly generated assertions such as (puppy, lay next to, chair) and (dogs, lay next to, pepperoni pizza) were rated as plausible (positives). 15.3% of annotations on our expanded tuples were positive. Overall, 36% of the labels in VAL and 37% of the labels in TEST are positives.

Tuple Illustration Interface

I collect abstract illustrations for all 213 relations in VAL. I get each relation illustrated by 20 different workers on AMT using the interface shown in Figure 5.2. Each worker is shown a *background* scene and asked to modify it to contain the relation of interest. I used living room scenes from (Antol et al. 2015) as background scenes, which were realistic scenes created by AMT workers using the same abstract scenes vocabulary as ours (Sec. 5.1.1). Priming workers with different background scenes helps increase the diversity in the visual illustrations of relations. For instance, when asked to create a scene depicting 'holding', a majority of workers might default to thinking of a person holding something while standing. But if they are primed with a scene where a woman is already sitting on a couch, then they might place a glass in her hand to make her hold the glass, resulting in a sitting person holding something. Workers are then instructed to indicate which clipart pieces in the scene correspond to the primary and secondary objects participating in the relation, and name them using as few words as possible.

To summarize, I collected 20 scenes depicting each of the 213 relations in VAL (4260 scenes total), along with annotations for the primary and secondary nouns and corresponding clipart objects participating in the relation. These form the set of TRAIN tuples that will be used to train the visual models of what tuples looks like. The VAL tuples will be used to

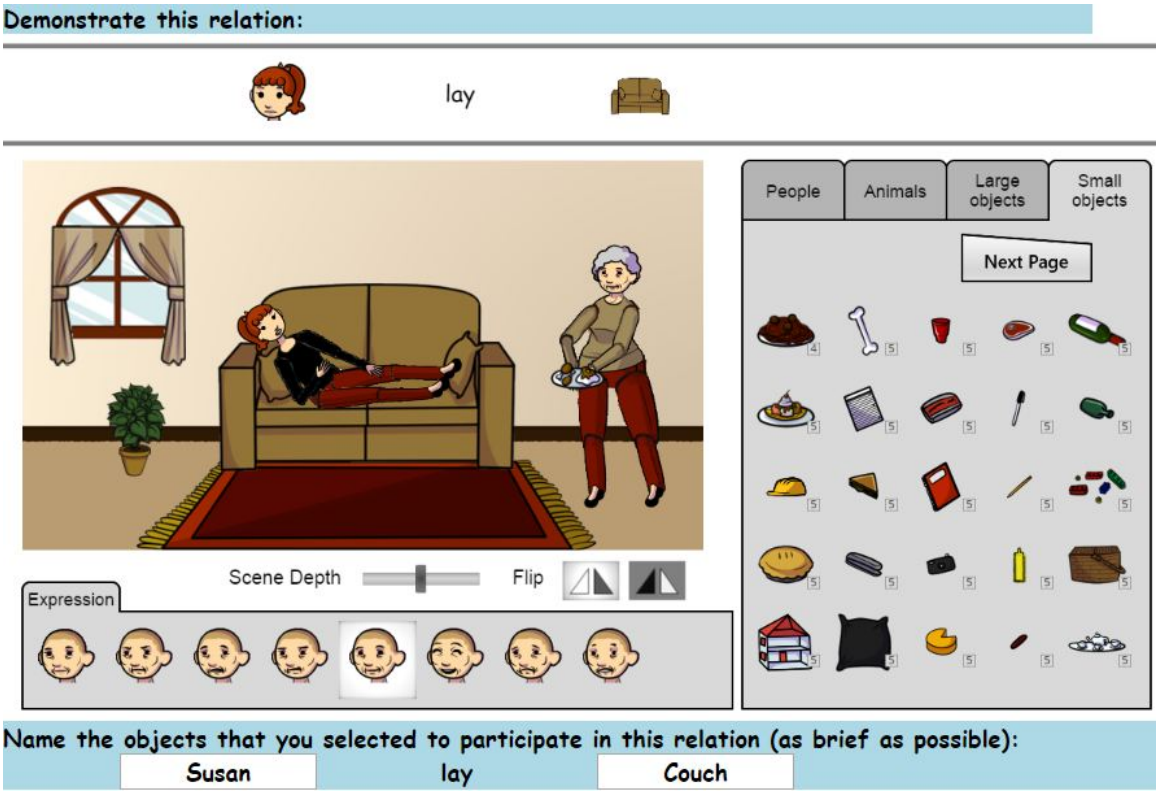


Figure 5.2: Our tuple illustration AMT interface.

learn how much visual alignment is weighted relative to the textual alignment. The TEST tuples will be used to evaluate the performance of our approach.

Note that I do not collect illustrations for each VAL *tuple* because tuples may contain nouns that our clipart library does not have. Instead, I collect illustrations for each of the VAL *relations*. Workers choose to depict these relations with plausible primary and second objects of their choice, providing an additional source of commonsense knowledge. Regardless, as will be evident in the next section, the model is capable of dealing with nouns and relations at test time that were not present during training.

5.1.2 Approach

I first describe the joint text and vision model, followed by a description of the training procedure.

Model

Let us start by laying out some notation. The model is given a commonsense assertion $t' = (t'_P, t'_R, t'_S)$ at test time, whose plausibility is to be evaluated. t'_P is the primary noun, t'_R is the relation, and t'_S is the secondary noun. For each abstract training scene created by AMT workers $i \in I$ we are given the primary and secondary clipart objects c^i_P and c^i_S, as well as a tuple $t^i = (t^i_P, t^i_R, t^i_S)$ containing the names of the primary and secondary objects (nouns), and the relation they participate in. Thus, a training instance i is represented by $\Omega^i = \{c^i_P, c^i_S, t^i\}$.

We can score the plausibility of test tuple t' using the following linear scoring function:

$$score(t') = \alpha \cdot f_{text}(t') + \beta \cdot f_{visual}(t') \tag{5.1}$$

Where α and β tradeoff the weights given to the text alignment score f_{text} and the vision alignment score f_{vision} respectively. The text and vision alignment scores estimate how well the test tuple t' aligns to all training instances – both textual (TRAIN tuples provided by AMT workers) and visual (training abstract scenes provided by AMT workers). Tuples which align well with known (previously seen and/or read) concepts are considered to be more plausible.

Vision and text alignment functions: Both our vision and text alignment functions have the following form:

$$f(t') = \frac{1}{|I|} \sum_{i \in I} \max(h(t', \Omega^i) - \delta, 0) \tag{5.2}$$

Where f can be either f_{text} or f_{vision}. The average goes over all training instances (i.e., abstract scenes with associated annotated tuples) in the training set. The activation of a training instance with respect to a test tuple is determined by h, which has different forms for vision and text. A ReLU (Rectified Linear Unit) function is applied to the activation

score offset by δ. I use a threshold of zero for the ReLU because the notion of negative plausibility evidence for a tuple is not intuitive. One can view Equation 5.2 as counting how many times a tuple was observed during training. The parameter δ is used to threshold the activation h to estimate counts. From here on I will refer to h as the alignment score (overloaded with f).

Text alignment score: The textual alignment score h_{text} between two tuples is a linear combination of similarities between the corresponding pairs of primary nouns, relations, and secondary nouns. These similarities are computed using dot products in the word2vec embedding space (Mikolov et al. 2013). For nouns or relations containing more than one word (e.g., "gather around" or "chair legs"), I average the word2vec vectors of each word to obtain a single vector.

Let $W(x)$ be the vector space embedding of a noun or relation x. The text alignment score is given as follows:

$$h_{text}(t', \Omega^i) = W(t'_P)^T \cdot W(t^i_P) + W(t'_R)^T \cdot W(t^i_R) + W(t'_S)^T \cdot W(t^i_S) \tag{5.3}$$

Where $\cdot$ denotes the cosine similarity between vectors.

Vision alignment score: The visual alignment score computes the alignment between (i) a given test tuple and (ii) the pair of clipart pieces selected by AMT workers as being the primary and secondary objects in a training instance i. It measures how well the pair of clipart pieces (c^i_P, c^i_S) depict the test tuple t'. If a test tuple finds support from a large number of visual instances, it is likely to be plausible. Note that we are measuring similarity between words and arrangements of clipart pieces. Consequently, this is a multimodal similarity function.

Given the pair of primary and secondary clipart pieces annotated in training instance Ω^i, I extract features as described in Section 5.1.4. I denote these extracted features as $u(c^i_P, c^i_S)$.

Using these visual features from the training instance Ω^i and text embeddings from test tuple t', I compute the following vision alignment score:

$$h_{vision}(t', \Omega^i) = u(c_P^i, c_S^i)^T A_P W(t'_P) + u(c_P^i, c_S^i)^T A_R W(t'_R) + u(c_P^i, c_S^i)^T A_S W(t'_S) \quad (5.4)$$

Where A_P, A_R, and A_S are alignment parameters to be learnt. The vision alignment score measures how well the t'_P, t'_R, and t'_S individually match the visual features $u(c_P^i, c_S^i)$ that describe a pair of clipart objects in training instance Ω_i. One can think of $u(c_P^i, c_S^i)A_P$, $u(c_P^i, c_S^i)A_R$, and $u(c_P^i, c_S^i)A_S$ as embeddings or projections from the vision space to the word2vec text space, such that a high dot product in word2vec space leads to high alignment, and subsequently a high plausibility score for plausible tuples. The embeddings are learnt separately for t'_P, t'_R and t'_S (as parameterized by A_P, A_R and A_S) because different visual features might be useful for aligning to the primary noun, relation, and secondary noun.

The parameters A_P, A_R, and A_S can also be thought of as grounding parameters. That is, given a word2vec vector W, we want to learn parameters to find the visual instantiation of W. $A_R W(t'_R)$ can be thought of as the visual instantiation of t'_R which captures what the interaction between two objects related by relation t'_R looks like. $A_P W(t'_P)$ and $A_S W(t'_S)$ can be thought of as identifying which clipart pieces and with what attributes correspond to nouns t'_P and t'_S. The model finds the visual grounding of t'_P, t'_R, and t'_S separately, and then measures similarity of the inferred grounding to the actual visual features observed in training instances. Thus, given a test tuple, I *hallucinate* a grounding for it and measure similarity of the hallucination with the training data. Note that these hallucinations are learnt discriminatively to help us align concepts in vision and text such that plausible tuples are scored highly.

5.1.3 Training

To learn the parameters A_P, A_R, A_S in our vision alignment scoring function (Equation 5.4), I consider the outer product space of the vectors u and W. I learn a linear SVM in this space to separate the training instances (tuples + corresponding abstract scenes, Section 5.1.1), from a set of negatives. Each negative instance is a tuple from our TRAIN set, paired with a random abstract scene from our training data. I sample three times as many negatives as positives. Overall this results in 4260 positives and 12780 negatives. Finally, the learnt vectors are reshaped to get A_P, A_R and A_S respectively. I learn the vision vs. text tradeoff parameters α and β (Equation 5.1) on the VAL set of tuples (Section 5.1.1). Recall that these include seed and expanded tuples, along with annotations indicating which tuples are plausible and which are not. I use the vision and text alignment scores as features and train a binary SVM to separate plausible tuples from implausible ones. The weights learnt by the SVM correspond to α and β. Finally, the parameter δ in Equation 5.2 is set using grid search on the VAL set to maximize the average precision (AP) of predicting a tuple as being plausible (positive) or not.

5.1.4 Experimental Setup

I will first describe the features extracted from the abstract scenes. I will then list the baselines we compare to.

Visual Features

As explained in Section 5.1.1, I have annotations indicating which pairs of objects (c_P, c_S) in an abstract scene participated in the corresponding annotated tuple. Using these objects and the remaining scene, I extract three kinds of features to describe the pair of objects (c_P, c_S): 1) Object Features 2) Interaction Features 3) Scene Features. These three together form the visual feature set. **Object Features** consist of the type (category, instance) of the object (Section 5.1.1), flip (left facing or right) of the object, absolute

location, attributes (for humans), and poses (for humans and animals). The absolute location feature is modeled using a Gaussian Mixture Model (GMM) with 9 components, learnt separately across five discrete depth levels, similar to (Zitnick, Parikh, and Vanderwende 2013). The GMM components are common across all objects, and are learnt using all objects present in all abstract scenes. Human attributes are age (5 discrete values), skin color (3 discrete values) and gender (2 discrete values). Animals have 5 discrete poses. Human pose features are constructed using keypoint locations. These include global, contact, and orientation features (Antol, Zitnick, and Parikh 2014). Global features measure the position of joints with respect to three gaussians placed on the head, torso, and feet respectively. Contact features place smaller gaussians at each joint and measure the positions of other joints with respect to each joint. Orientation features measure the joint angles between connected keypoints. **Interaction Features** encode the relative locations of the two objects participating in the relation, normalized for the flip and depth of the first object. This results in the relative location features being asymmetric. I compute the relative location of the primary object relative to the secondary object and vice versa. Relative locations are encoded using a 24 component GMM (similar to (Zitnick, Parikh, and Vanderwende 2013)). **Scene Features** indicate which types (category, instance) of objects (other than c_P and c_S) are present in the scene. Overall, there are 493 object features each for the primary and secondary objects, 48 interaction features, and 188 global features, resulting in a visual feature vector of dimension 1222.

Baselines

I experimented with a variety of strong baselines that use text information alone. They help evaluate how much complementary information vision adds, and if this additional information can be obtained simply from additional or different kinds of text (e.g., generic vs. visual text).

- **WikiEmbedding**: The first baseline uses the f_{text} part of the model (Equation 5.1)

alone. It uses word2vec trained on generic Wikipedia text.

- **COCOEmbedding**: Our next baseline also uses the f_{text} part of the model (Equation 5.1) alone, but uses word2vec trained on visual text (>400k captions in the COCO training dataset).
- **ValText**: Recall that both the TEST and VAL tuples were extracted from captions describing COCO images. Our next baseline computes the plausibility of a test tuple by counting how often that tuple occurred in VAL. This helps assess the overlap between our TEST and VAL tuples (recall: no images are shared between TEST and VAL). Note that the above two baselines, WikiEmbedding and COCOEmbedding, can be thought of as ValText but by using soft similarities (in word2vec space) rather than using counts based on exact matches.
- **LargeVisualText**: The next baseline is a stronger version of ValText. Instead of using just our VAL tuples to evaluate the plausibility of a test tuple, it extracts tuples from a large corpus of text describing images (>400k captions in the MS COCO training dataset which are not in the test set (Section 5.1.1)). This gives us a set of 91K assertions. At test time, one can check how many times the test assertion occurred in this set, and use that count as the plausibility score of the test tuple.
- **BigGenericText (Bing)**: In this baseline, I evaluate the performance of assessing the plausibility of tuple $t' = (t'_P, t'_R, t'_S)$ in the test set using all the text on the web. Specifically, I query the Bing[3] search API and compute the log-frequencies of t'_P, t'_R, t'_S as well as t'. I train an SVM on these four features to separate plausible tuples in our VAL set from implausible tuples, and use this SVM at test time to compute the plausibility score of a test tuple.

[3]http://www.bing.com/

Evaluation

Recall that I collected 10 human judgements for the plausibility of each test tuple (Section 5.1.1). Next, I counted the number of subjects who thought the tuple was plausible ($count_+$). I also counted the number of subjects who thought the tuple was not plausible ($count_-$). $count_+$ + $count_-$ need not be 10 because subjects were allowed to marked tuples as "does not make sense". These scores are then combined into a single $score = count_+ - count_-$. I then threshold these scores at 0 to get the set of positive and negative human (ground truth) labels. That is, a tuple is considered to be plausible if more people thought it is plausible than not. The method as well as the baselines produce a score for the plausibility of each tuple in the TEST set. These scores are thresholded and compared to the human labels to compute average precision (AP). I also rank tuples based on their predicted plausibility scores and human plausibility scores ($score = count_+ - count_-$). These rankings are compared using a rank correlation, which forms the second evaluation metric.

5.1.5 Results

I begin by comparing the text-based baseline models. I then demonstrate the advantage of using vision and text jointly, over using text alone or vision alone. After that, I will showcase qualitative results. Finally, I comment on the potential our approach has to enrich existing knowledge bases.

Different Text Models

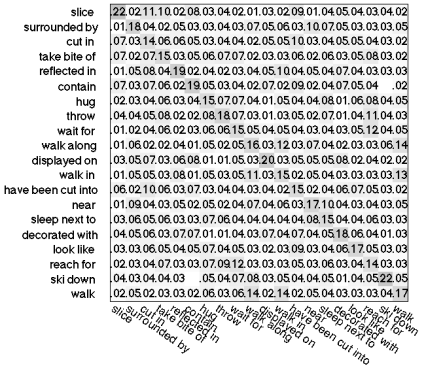
Of all the text-alone baselines (Table. 5.1), I find that BigGenericText (Bing) does the worst, likely because it suffers heavily from the reporting bias on the web. The LargeVisualText baseline does better than Bing, presumably because the captions in COCO describe what is seen in the images which may often be mundane details depicted in the image, and aligns well with the source of the tuples (visual text). ValText performs worse than LargeVisualText

Table 5.1: Performance of different text based methods on common sense assertion scoring.

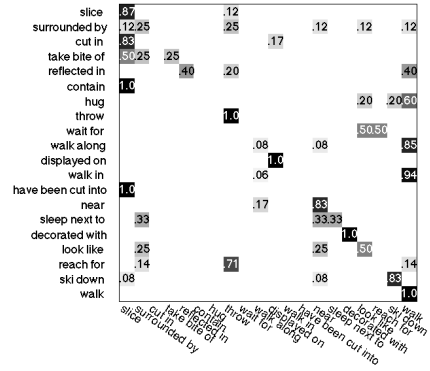
Approach	Test Performance	
	AP	Rank Correlation × 100
WikiEmbedding	68.4	41.7
COCOEmbedding	72.2	49.0
ValText	53.0	31.0
LargeVisualText	58.0	37.6
BigGenericText (Bing)	44.6	20.3

Table 5.2: Text+ vision outperforms text alone on commonsense assertion scoring.

Approach	Test Performance	
	AP	Rank Correlation × 100
Text (COCOEmbedding) + Vision	73.6	50.0
Vision Only	68.7	45.3
Text (COCOEmbedding) Only	72.2	49.0



(a) Textual similarity between relations



(b) Visual similarity between relations

Figure 5.3: Visual and textual similarities are qualitatively different, and capture complimentary signals for modeing common sense.

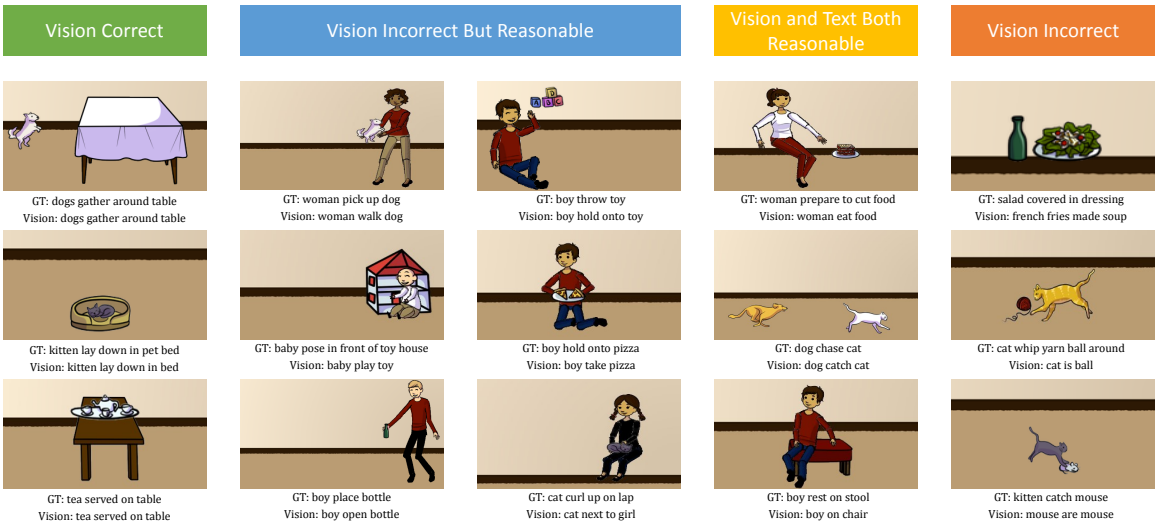


Figure 5.4: Qualitative examples demonstrating visual similarity between tuples.

because ValText uses less data. But adding soft similarities using word2vec embeddings (WikiEmbedding and COCOEmbedding) significantly improves performance (15.4 and 19.2 in absolute AP). COCOEmbedding performs the best among all text-alone baselines, and is what I will use as the "text only" model moving forward.

Joint Text + Vision Model

I compare the performance of text + vision, vision alone, and text alone in Table. 5.2. We can observe that text + vision performs better than text alone and vision alone by 1.4% and 4.9% AP respectively. In terms of rank correlation, text + vision provides an improvement of 1.0 over text alone. Overall, vision and text provide complementary sources of common sense.

Qualitative Results

I first visualize relation similarity matrices for text and vision alone (Figure 5.3). Each entry in the text matrix is the word2vec similarity between the relations specified in the corresponding row and columns. Each row is normalized to sum to 1. For vision, each entry in the matrix in the proportion of images depicting a relation (row) whose embeddings

– after being transformed by A_R – are most similar to the word2vec representation of another relation (column). This illustrates what the visual alignment function has learnt. We randomly sample a subset of 20 relations for visualization purposes. One can clearly see that the two matrices are qualitatively different and complementary. For instance, visual cues tell us that the relations like "sleep next to" and "surrounded by" are similar.

In Figure 5.4 I show several scenes created by AMT workers. Note that for clarity I only show the primary and secondary objects as identified by workers, but our approach uses all objects present in the scene. For each scene, I show the "GT" tuple provided by workers, as well as the "Vision only" tuple. This is computed by embedding the scene using our learnt A_P, A_R, and A_S into the word2vec space and identifying the nouns and relations that are most similar. The left most column shows scenes where the visual prediction matches the GT. The next column shows scenes where the visual prediction is incorrect, but reasonable (even desirable) and would not be captured by text. Consider (boy, hold onto, pizza) and (boy, take, pizza) whose similarity would be difficult to capture via text. The next column shows examples where the tuples are visually as well as textually similar. The last column shows failure cases where the visual prediction is unreasonable.

Enriching Knowledge Bases

ConceptNet (Speer and Havasi 2012) contains commonsense knowledge contributed by volunteers. It represents concepts with nodes and relations as edges between them. Out of our 213 VAL relations, only one relation ("made of") currently exists in ConceptNet. Thus, this approach can add many visual commonsense relations to ConceptNet, and boost its recall.

5.2 Visual-word2vec: Learning Visually Grounded Word Embeddings Using Abstract Scenes

Next, I will discuss an approach to learn grounded word embeddings using aligned image and text data. In practice I learn these embeddings using the commonsense tuples from Sec. 5.1, and from other previous work (Lin and Parikh 2015), and apply them to tasks like commonsense assertion classification (from Sec. 5.1), visual paraphrasing (Lin and Parikh 2015), and text-based image retrieval.

My approach considers visual cues from abstract scenes as context for words. Given a set of words and associated abstract scenes, I first cluster the scenes in a rich semantic feature space capturing the presence and locations of objects, pose, expressions, gaze, age of people, *etc*. Note that these features can be trivially extracted from abstract scenes. Using these features helps to capture fine-grained notions of semantic relatedness (Fig. 5.7). I then train to predict the cluster membership from pre-initialized word embeddings. The idea is to bring embeddings for words with similar visual instantiations closer, and push words with different visual instantiations farther (Fig. 1.4). The word embeddings are initialized with word2vec (Mikolov et al. 2013). The clusters thus act as surrogate classes. Note that each surrogate class may have images belonging to concepts which are different in text, but are visually similar. Since I predict the visual clusters as context given a set of input words, the model can be viewed as a multi-modal extension of the continuous bag of words (CBOW) (Mikolov et al. 2013) word2vec model.

5.2.1 Approach

Recall that the `vis-w2v` model grounds word embeddings into vision by treating vision as context. I first detail our inputs. I then discuss our `vis-w2v` model. Next, I'll describe the clustering procedure to get surrogate semantic labels, which are used as visual context by the model. I will then describe how word-embeddings are initialized. Finally, I will draw

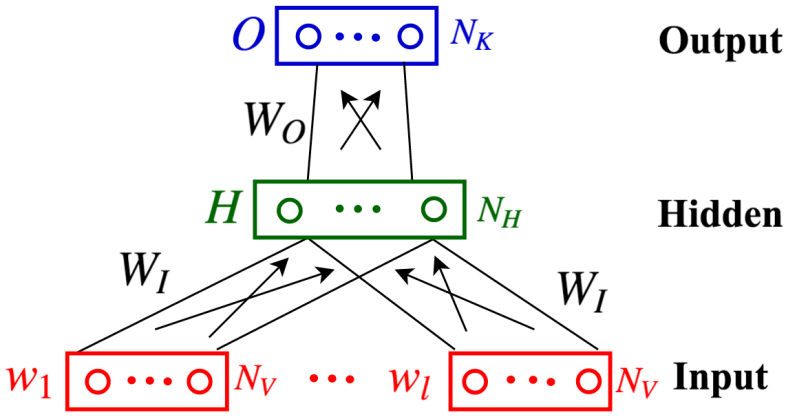


Figure 5.5: Proposed `vis-w2v` model. The input layer (red) has multiple one-hot word encodings. These are connected to the hidden layer with the projection matrix W_I, *i.e.*, all the inputs share the same weights. It is finally connected to the output layer via W_O. Model predicts the visual context O given the text input $S_w = \{w_l\}$.

connections to word2vec (`w2v`) models.

Input: We are given a set of pairs of visual scenes and associated text $D = \{(v, w)\}_d$ in order to train `vis-w2v`. Here v refers to the image features and w refers to the set of words associated with the image. At each step of training, we select a window $S_w \subseteq w$ to train the model.

Model: The `vis-w2v` model (Fig. 5.5) is a neural network that accepts as input a set of words S_w and a visual feature instance v. Each of the words $w_i \in S_w$ is represented via a one-hot encoding. A one-hot encoding enumerates over the set of words in a vocabulary (of size N_V) and places a 1 at the index corresponding to the given word. This one-hot encoded input is transformed using a projection matrix W_I of size $N_V \times N_H$ that connects the input layer to the hidden layer, where the hidden layer has a dimension of N_H. Intuitively, N_H decides the capacity of the representation. Consider an input one-hot encoded word w_i whose j^{th} index is set to 1. Since w_i is one-hot encoded, the hidden activation for this word (H_{w_i}) is a row in the weight matrix W_I^j, *i.e.*, $H_{w_i} = W_I^j$. The resultant hidden activation H would then be the average of individual hidden activations H_{w_i} as W_I is shared among all

the words S_w, *i.e.*,:

$$H = \frac{1}{|S_w|} \sum_{w_i \in S_w \subseteq w} H_{w_i} \tag{5.5}$$

Given the hidden activation H, we multiply it with an output weight matrix W_O of size $N_H \times N_K$, where N_K is the number of output classes. The output class (described next) is a discrete-valued function of the visual features $G(v)$ (more details in next paragraph). We normalize the output activations $O = H \times W_O$ to form a distribution using the softmax function. Given the softmax outputs, we minimize the negative log-likelihood of the correct class conditioned on the input words:

$$\min_{W_I, W_O} -\log P(G(v)|S_w, W_I, W_O) \tag{5.6}$$

We optimize for this objective using stochastic gradient descent (SGD) with a learning rate of 0.01.

Output Classes: As mentioned in the previous section, the target classes for the neural network are a function $G(\cdot)$ of the visual features. What would be a good choice for G? Recall that our aim is to recover an embedding for words that respects similarities in visual instantiations of words (Fig. 1.4). To capture this visual similarity, we model $G : v \rightarrow \{1, \cdots, N_K\}$ as a grouping function[4]. In practice, this function is learnt offline using clustering with K-means. That is, the outputs from clustering are the surrogate class labels used in `vis-w2v` training. Since we want our embeddings to reason about fine-grained visual grounding (*e.g.*"stares at" and "eats"), we cluster in the abstract scenes feature space from the previous section (Sec. 5.1). See Fig. 5.7 for an illustration of what clustering captures. The parameter N_K in K-means modulates the granularity at which we reason

[4]Alternatively, one could regress directly to the feature values v. However, we found that the regression objective hurts performance.

about visual grounding.

Initialization: We initialize the projection matrix parameters W_I with those from training `w2v` on large text corpora. The hidden-to-output layer parameters are initialized randomly. Using `w2v` is advantageous for us in two ways: i) `w2v` embeddings have been shown to capture rich semantics and generalize to a large number of tasks in text. Thus, they provide an excellent starting point to finetune the embeddings to account for visual similarity as well. ii) Training on a large corpus gives us good coverage in terms of the vocabulary. Further, since the gradients during backpropagation only affect parameters/embeddings for words seen during training, one can view `vis-w2v` as augmenting `w2v` with visual information when available. In other words, we retain the rich amount of non-visual information already present in it[5]. Indeed, we find that the random initialization does not perform as well as initialization with `w2v` when training `vis-w2v`.

Design Choices: Our model (Sec. 5.2.1) admits choices of w in a variety of forms such as full sentences or tuples of the form (Primary Object, Relation, Secondary Object). The exact choice of w is made depending upon on what is natural for the task of interest. For instance, for common sense assertion classification and text-based image retrieval, w is a phrase from a tuple, while for visual paraphrasing w is a sentence. Given w, the choice of S_w is also a design parameter tweaked depending upon the task. It could include all of w (*e.g.*, when learning from a phrase in the tuple) or a subset of the words (*e.g.*, when learning from an n-gram context-window in a sentence). While the model itself is task agnostic, and only needs access to the words and visual context during training, the validation and test performances are calculated using the `vis-w2v` embeddings on a specific task of interest (Sec. 5.2.2). This is used to choose the hyperparameters N_K and N_H.

Connections to `w2v`: Our model can be seen as a multi-modal extension of the continuous

[5]We verified empirically that this does not cause *calibration* issues. Specifically, given a pair of words where one word was refined using visual information but the other was not (unseen during training), using `vis-w2v` for the former and `w2v` for the latter when computing similarities between the two outperforms using `w2v` for both.

bag of words (CBOW) w2v models. The CBOW w2v objective maximizes the likelihood $P(w|S_w, W_I, W_O)$ for a word w and its context S_w. On the other hand, we maximize the likelihood of the visual context given a set of words S_w (Eqn. 5.6).

5.2.2 Applications

We compare vis-w2v and w2v on the tasks of common sense assertion classification (Sec. 5.2.2), visual paraphrasing (Sec. 5.2.2), and text-based image retrieval (Sec. 5.2.2). We give details of each task and the associated datasets below.

Common Sense Assertion Classification

We study the relevance of vis-w2v to the common sense (CS) assertion classification task introduced in the previous section. Given common sense tuples of the form (primary object or t_P, relation or t_R, secondary object or t_S) *e.g.* (boy, eats, cake), the task is to classify it as plausible or not. The CS dataset contains $14,332$ TEST assertions (spanning 203 relations) out of which 37% are plausible, as indicated by human annotations. These TEST assertions are extracted from the MS COCO dataset (Lin et al. 2014), which contains real images and captions. Evaluating on this dataset allows us to demonstrate that visual grounding learnt from the abstract world generalizes to the real world. The previous section approaches the task by constructing a multi-modal similarity function between TEST assertions whose plausibility is to be evaluated, and TRAIN assertions that are known to be plausible. The TRAIN dataset also contains 4260 abstract scenes made from clipart depicting 213 relations between various objects (20 scenes per relation). Each scene is annotated with one tuple that names the primary object, relation, and secondary object depicted in the scene. Abstract scene features (from Sec. 5.1) describing the interaction between objects such as relative location, pose, absolute location, *etc*. are used for learning vis-w2v. We use the VAL set from the previous section ($14,548$ assertions) to pick the hyperparameters. Since the dataset contains tuples of the form (t_P, t_R, t_S), we explore

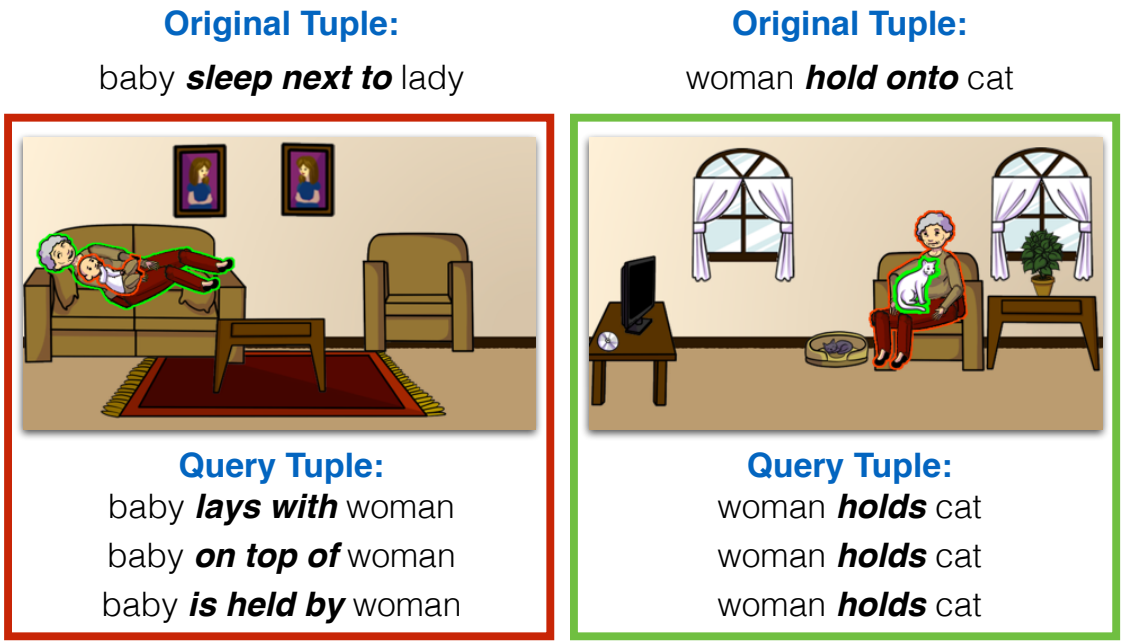


Figure 5.6: Examples tuples collected for the text-based image retrieval task. Notice that multiple relations can have the same visual instantiation (left).

learning `vis-w2v` with **separate** models for each, and a **shared** model irrespective of the word being t_P, t_R, or t_S.

Visual Paraphrasing

Visual paraphrasing (VP), introduced by Lin and Parikh (Lin and Parikh 2015) is the task of determining if a pair of descriptions describes the same scene or two different scenes. The dataset introduced by (Lin and Parikh 2015) contains $30,600$ pairs of descriptions, of which a third are positive (describe the same scene) and the rest are negatives. The TRAIN dataset contains $24,000$ VP pairs whereas the TEST dataset contains $6,060$ VP pairs. Each description contains three sentences. We use scenes and descriptions from Zitnick *et.al* (Zitnick, Parikh, and Vanderwende 2013) to train `vis-w2v` models, similar to Lin and Parikh. The abstract scene feature set from (Zitnick, Parikh, and Vanderwende 2013) captures occurrence of objects, person attributes (expression, gaze, and pose), absolute spatial location and co-occurrence of objects, relative spatial location between pairs of objects, and depth ordering (3 discrete depths), relative depth and flip. We withhold a

set of 1000 pairs (333 positive and 667 negative) from TRAIN to form a VAL set to pick hyperparameters. Thus, our VP TRAIN set has $23,000$ pairs.

Text-based Image Retrieval

In order to verify if our model has learnt the visual grounding of concepts, we study the task of text-based image retrieval. Given a query tuple, the task is to retrieve the image of interest by matching the query and ground truth tuples describing the images using word embeddings. For this task, we study the generalization of `vis-w2v` embeddings learnt for the common sense (CS) task, *i.e.*, there is no training involved. We augment the common sense (CS) dataset (Sec. 5.2.2) to collect three query tuples for each of the original 4260 CS TRAIN scenes. Each scene in the CS TRAIN dataset has annotations for which objects in the scene are the primary and secondary objects in the ground truth tuples. We highlight the primary and secondary objects in the scene and ask workers on AMT to name the primary, secondary objects, and the relation depicted by the interaction between them. Some examples can be seen in Fig. 5.6. Interestingly, some scenes elicit diverse tuples whereas others tend to be more constrained. This is related to the notion of Image Specificity (Jas and Parikh 2015). Note that the workers do not see the original (ground truth) tuple written for the scene from the CS TRAIN dataset. We use the collected tuples as queries for performing the retrieval task. Note that the queries used at test time were never used for training `vis-w2v`.

5.2.3 Experimental Setup

We now explain our experimental setup. We first explain how we use our `vis-w2v` or baseline `w2v` (word2vec) model for the three tasks described above: common sense (CS), visual paraphrasing (VP), and text-based image retrieval. We also provide evaluation details. We then list the baselines we compare to for each task and discuss some design choices. For all the tasks, we preprocess raw text by tokenizing using the NLTK toolkit (Loper and Bird 2002). We implement `vis-w2v` as an extension of the Google C implementation of

word2vec[6].

Common Sense Assertion Classification

The task in common sense assertion classification (Sec. 5.1) is to compute the plausibility of a test assertion based on its similarity to a set of tuples ($\Omega = \{t^i\}_{i=1}^{I}$) known to be plausible. Given a tuple $t' =$ `(Primary Object` t'_P`, Relation` t'_R`, Secondary Object` t'_S`)` and a training instance t^i, the plausibility scores are computed as follows:

$$h(t', t^i) = W_P(t'_P)^T W_P(t^i_P) + W_R(t'_R)^T W_R(t^i_R) + W_S(t'_S)^T W_S(t^i_S) \tag{5.7}$$

where W_P, W_R, W_S represent the corresponding word embedding spaces. The final text score is given as follows:

$$f(t') = \frac{1}{|I|} \sum_{i \in I} \max(h(t', t^i) - \delta, 0) \tag{5.8}$$

where i sums over the entire set of training tuples. I reuse the value of δ used in Sec. 5.1 for these experiments.

Similar to Sec. 5.1 I share embedding parameters across t_P, t_R, t_S in the text based model in one setting. That is, $W_P = W_R = W_S$ (let us call this the **shared** model). And when W_P, W_R, W_S are learnt independently for (t_P, t_R, t_S), let us call it the **separate** model.

The approach in Sec. 5.1 also has a visual similarity function that combines text and abstract scenes that is used along with this text-based similarity. Here I use the text-based approach for evaluating both `vis-w2v` and baseline `w2v`. However, I also report results including the visual similarity function along with text similarity from `vis-w2v`. In line with Sec. 5.1, I also evaluate the results here using average precision (AP) as a performance metric.

[6]https://code.google.com/p/word2vec/

Visual Paraphrasing

In the visual paraphrasing task (Sec. 5.2.2), we are given a pair of descriptions at test time. We need to assign a score to each pair indicating how likely they are to be paraphrases, *i.e.*, describing the same scene. Following (Lin and Parikh 2015) we average word embeddings (`vis-w2v` or `w2v`) for the sentences and plug them into their text-based scoring function. This scoring function combines term frequency, word co-occurrence statistics and averaged word embeddings to assess the final paraphrasing score. The results are evaluated using average precision (AP) as the metric. While training both `vis-w2v` and `w2v` for the task, we append the sentences from the train set of (Lin and Parikh 2015) to the original word embedding training corpus to handle vocabulary overlap issues.

Text-based Image Retrieval

I compare `w2v` and `vis-w2v` on the task of text-based image retrieval (Sec. 5.2.2). The task involves retrieving the target image from an image database, for a query tuple. Each image in the database has an associated ground truth tuple describing it. We use these to rank images by computing similarity with the query tuple. Given tuples of the form (t_P, t_R, t_S), I average the vector embeddings for all words in t_P, t_R, t_S. I then explore **separate** and **shared** models just as we did for common sense assertion classification. In the **separate** model, I first compute the cosine similarity between the query and the ground truth for t_P, t_R, t_S separately and average the three similarities. In the **shared** model, I average the word embeddings for t_P, t_R, t_S for query and ground truth and then compute the cosine similarity between the averaged embeddings. The similarity scores are then used to rank the images in the database for the query. I use standard metrics for retrieval tasks to evaluate: `Recall@1` (`R@1`), `Recall@5` (`R@5`), `Recall@10` (`R@10`) and median rank (`med R`) of target image in the returned result.

Table 5.3: Performance on the common sense task proposed in Sec. 5.1

Approach	common sense AP (%)
vis-w2v-wiki (shared)	72.2
vis-w2v-wiki (separate)	74.2
vis-w2v COCO (shared) + vision	74.2
vis-w2v COCO (shared)	74.5
vis-w2v COCO (separate)	**74.8**
vis-w2v COCO (separate) + vision	**75.2**
w2v-wiki (from Sec. 5.1)	68.4
w2v COCO (from Sec. 5.1)	72.2
w2v COCO + vision (from Sec. 5.1)	73.6

Baselines

We describe some baselines in this subsection. In general, we consider two kinds of w2v models: those learnt from generic text, *e.g.*, Wikipedia (w2v-wiki) and those learnt from visual text, *e.g.*, COCO (w2v COCO), *i.e.*, text describing images. As noted in the previous section, embeddings learnt from visual text typically contain more visual information. vis-w2v-wiki are vis-w2v embeddings learnt using w2v-wiki as an initialization to the projection matrix, while vis-w2v COCO are the vis-w2v embeddings learnt using w2v COCO as the initialization. In all settings, we are interested in studying the performance gains on using vis-w2v over w2v. Although our training procedure itself is task agnostic, we train separately on the common sense (CS) and the visual paraphrasing (VP) datasets. We study generalization of the embeddings learnt for the CS task on the text-based image retrieval task. Additional design choices pertaining to each task are discussed in Sec. 5.2.1.

5.2.4 Results

We present results on common sense (CS), visual paraphrasing (VP), and text-based image retrieval tasks. We compare our approach to various baselines as explained in Sec. 5.2.3 for each application. Finally, we train our model using real images instead of abstract scenes, and analyze differences.

Common Sense Assertion Classification

I first present the results on the common sense assertion classification task (Sec. 5.1). I report numbers with a fixed hidden layer size, $N_H = 200$ (to be comparable to Sec. 5.1) in Table. 5.3. I use $N_K = 25$, which gives the best performance on validation. I handle tuple elements, t_P, t_R or t_S, with more than one word by placing each word in a separate window (*i.e.*$|S_w| = 1$). For instance, the element "lay next to" is trained by predicting the associated visual context thrice with "lay", "next" and "to" as inputs. Overall, we find an increase of 2.6% with `vis-w2v` COCO (separate) model over the `w2v` COCO model used in Sec. 5.1. We see larger gains (5.8%) with `vis-w2v-wiki` over `w2v-wiki`. Interestingly, since the tuples in the common sense task are extracted from the MS COCO (Lin et al. 2014) dataset, this is an instance where `vis-w2v` (learnt from abstract scenes) generalizes to text describing real images.

The `vis-w2v`COCO (both shared and separate) embeddings outperform the joint `w2v`COCO + vision model from Sec. 5.1 that reasons about visual features for a given test tuple, which we do not do here. Note that both models use the same training and validation data, which suggests that the `vis-w2v` model captures the grounding better than the previous multi-modal text + visual similarity model. Finally, we sweep for the best value of N_H for the validation set and find that `vis-w2v` COCO (separate) gets the best AP of 75.4% on TEST with $N_H = 50$. This is our best performance on this task.

Separate vs. Shared: We next compare the performance when using the **separate** and **shared** `vis-w2v` models. We find that `vis-w2v` COCO (separate) does better than `vis-w2v` COCO (shared) (74.8% vs. 74.5%), presumably because the embeddings can specialize to the semantic roles words play when participating in t_P, t_R or t_S. In terms of **shared** models alone, `vis-w2v` COCO (shared) achieves a gain in performance of 2.3% over the `w2v` COCO model from Sec. 5.1, where the textual models are all shared.

What Does Clustering Capture? We next visualize the semantic relatedness captured by

Figure 5.7: Visualization of the clustering used to supervise `vis-w2v` training. Relations that co-occur more often in the same cluster appear bigger than others. Observe how semantically close relations co-occur the most, *e.g.*, `eat`, `drink`, `chew on` for the relation `enjoy`.

clustering in the abstract scenes feature space (Fig. 5.7). Recall that clustering gives us surrogate labels to train `vis-w2v`. For the visualization, we pick a relation and display other relations that co-occur the most with it in the same cluster. Interestingly, words like "prepare to cut", "hold", "give" occur often with "stare at". Thus, we discover the fact that when we "prepare to cut" something, we also tend to "stare at" it. Reasoning about such notions of semantic relatedness using purely textual cues would be prohibitively difficult.

Visual Paraphrasing

I next describe the results on the Visual Paraphrasing (VP) task (Lin and Parikh 2015). The task is to determine if a pair of descriptions are describing the same scene. Each description has three sentences. Table. 5.4 summarizes the results and compares performance to `w2v`. I vary the size of the context window S_w and check performance on the VAL set. One can obtain best results with the entire description as the context window S_w, $N_H = 200$,

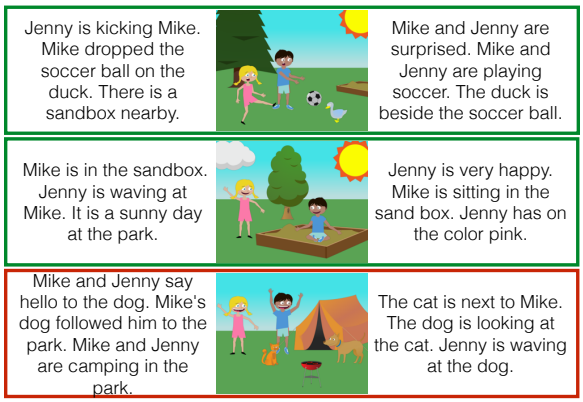


Figure 5.8: The visual paraphrasing task is to identify if two textual descriptions are paraphrases of each other. Shown above are three positive instances, *i.e.*, the descriptions (left, right) actually talk about the same scene (center, shown for illustration, not avaliable as input). Green boxes show two cases where `vis-w2v` correctly predicts and `w2v` does not, while red box shows the case where both `vis-w2v` and `w2v` predict incorrectly. Note that the red instance is tough as the textual descriptions do not intuitively seem to be talking about the same scene, even for a human reader.

and $N_K = 100$. The `vis-w2v` models give an improvement of 0.7% on both `w2v-wiki` and `w2v`COCO respectively. In comparison to `w2v-wiki` approach from (Lin and Parikh 2015), we get a larger gain of 1.2% with the `vis-w2v` COCO embeddings[7]. Lin and Parikh (Lin and Parikh 2015) imagine the visual scene corresponding to text to solve the task. Their combined text + imagination model performs 0.2% better (95.5%) than our model. Note that our approach does not have the additional expensive step of generating an imagined visual scene for each instance at test time. Qualitative examples of success and failure cases are shown in Fig. 5.8.

Window Size: Since the VP task is on multi-sentence descriptions, it gives us an opportunity to study how size of the window (S_w) used in training affects performance. We evaluate the gains obtained by using window sizes of entire description, single sentence, 5 words, and single word respectively. We find that description level windows and sentence level windows give equal gains. However, performance tapers off as we reduce the context to 5 words (0.6% gain) and a single word (0.1% gain). This is intuitive, since VP requires

[7]Our implementation of (Lin and Parikh 2015) performs 0.3% higher than that reported in (Lin and Parikh 2015).

Table 5.4: Performance on visual paraphrasing task of (Lin and Parikh 2015).

Approach	Visual Paraphrasing AP (%)
w2v-wiki (from (Lin and Parikh 2015))	94.1
w2v-wiki	94.4
w2v COCO	94.6
vis-w2v-wiki	95.1
vis-w2v COCO	**95.3**

Table 5.5: Performance on text-based image retrieval. R@x: **higher** is better, medR: **lower** is better

Approach	R@1 (%)	R@5 (%)	R@10 (%)	med R
w2v-wiki	14.6	34.4	45.4	13
w2v COCO	15.3	35.2	47.6	11
vis-w2v-wiki (shared)	15.5	37.2	49.3	**10**
vis-w2v COCO (shared)	**15.7**	**37.7**	47.6	**10**
vis-w2v-wiki (separate)	14.0	32.7	43.5	15
vis-w2v COCO (separate)	15.4	37.6	**49.5**	**10**

us to reason about entire descriptions to determine paraphrases. Further, since the visual features in this dataset are scene level (and not about isolated interactions between objects), the signal in the hidden layer is stronger when an entire sentence is used.

Text-based Image Retrieval

We next present results on the text-based image retrieval task (Sec. 5.2.2). This task requires visual grounding as the query and the ground truth tuple can often be different by textual similarity, but could refer to the same scene (Fig. 5.6). As explained in Sec. 5.2.2, we study generalization of the embeddings learnt during the commonsense experiments to this task. Table. 5.5 presents our results. Note that vis-w2v here refers to the embeddings learnt using the CS dataset. We find that the best performing models are vis-w2v-wiki (shared) (as per R@1, R@5, medR) and vis-w2v COCO (separate) (as per R@10, medR). These get Recall@10 scores of ≈49.5% whereas the baseline w2v-wiki and w2v COCO embeddings give scores of 45.4% and 47.6%, respectively.

Real Image Experiment

Finally, I test the vis-w2v approach with real images on the commonsense task, to evaluate the need to learn fine-grained visual grounding via abstract scenes. Thus, instead of semantic features from abstract scenes, we obtain surrogate labels by clustering real images from the MS COCO dataset using fc7 features from the VGG-16 (Simonyan and Zisserman 2015) CNN. I cross validate to find the best number of clusters and hidden units. I perform real image experiments in two settings: 1) using all of the MS COCO dataset after removing the images whose tuples are in the CS TEST set in Sec. 5.1. This gives us a collection of $\approx$ 76K images to learn vis-w2v. COCO dataset has a collection of 5 captions for each image. I use all these five captions with sentence level context[8] windows to learn vis-w2v80K. 2) I create a real image dataset by collecting 20 real images from MS COCO and their corresponding tuples, randomly selected for each of 213 relations from the VAL set (Sec. 5.2.3). Analogous to the CS TRAIN set containing abstract scenes, this gives us a dataset of 4260 real images along with an associated tuple, depicting the 213 CS VAL relations. I refer to this model as vis-w2v4K.

We report the gains in performance over w2v baselines in both scenario 1) and 2) for the common sense task. We find that using real images gives a best-case performance of 73.7% starting from w2v COCO for vis-w2v80K (as compared to 74.8% using CS TRAIN abstract scenes). For vis-w2v 4K COCO, the performance on the validation actually goes down during training. If we train vis-w2v4K starting with generic text based w2v-wiki, we get a performance of 70.8% (as compared to 74.2% using CS TRAIN abstract scenes). This shows that abstract scenes are better at visual grounding as compared to real images, due to their rich semantic features.

[8]I experimented with other choices but found this works best.

5.2.5 Discussion

Antol *et.al* (Antol, Zitnick, and Parikh 2014) have studied generalization of classification models learnt on abstract scenes to real images. The idea is to transfer fine-grained concepts that are easier to learn in the fully-annotated abstract domain to tasks in the real domain. Our work can also be seen as a method of studying generalization. One can view `vis-w2v` as a way to transfer knowledge learnt in the abstract domain to the real domain, via text embeddings (which are shared across the abstract and real domains). Our results on commonsense assertion classification show encouraging preliminary evidence of this.

We next discuss some considerations in the design of the model. A possible design choice when learning embeddings could have been to construct a triplet loss function, where the similarity between a tuple and a pair of visual instances can be specified. That is, given a textual instance A, and two images B and C (where A describes B, and not C), one could construct a loss that enforces $sim(A, B) > sim(A, C)$, and learn joint embeddings for words and images. However, since we want to learn hidden semantic relatedness (*e.g.* "eats", "stares at"), there is no explicit supervision available at train time on which images and words should be related. Although the visual scenes and associated text inherently provide information about related words, they do not capture the unrelatedness between words, *i.e.*, we do not have negatives to help us learn the semantics.

We can also understand `vis-w2v` in terms of data augmentation. With infinite text data describing scenes, distributional statistics captured by `w2v` would reflect all possible visual patterns as well. In this sense, there is nothing special about the visual grounding. The additional modality helps to learn complimentary concepts while making efficient use of data. Thus, the visual grounding can be seen as augmenting the amount of textual data.

CHAPTER 6

VISUAL IMAGINATION

In this chapter, our goal will be to study how to create models of visual imagination, which capture the appropriate denotation of concepts. One can motivate certain desiderata for visual imagination by considering the following two-party communication game: a speaker thinks of a visual concept C, such as "men with black hair", and then generates a description $\mathbf{y}$ of this concept, which she sends to a listener; the listener interprets the description $\mathbf{y}$, by creating an internal representation $\mathbf{z}$, which captures its "meaning". We can think of $\mathbf{z}$ as representing a set of "mental images" which depict the concept C. To test whether the listener has correctly "understood" the concept, we ask him to draw a set of real images $\mathcal{S} = \{\mathbf{x}_s : s = 1 : S\}$, which depict the concept C. He then sends these back to the speaker, who checks to see if the images correctly match the concept C. I call this process *visually grounded imagination*.

In this chapter, I will represent concept descriptions in terms of a fixed length vector of discrete attributes $\mathcal{A}$. This will allows us to specify an exponentially large set of concepts using a compact, combinatorial representation. In particular, by specifying different subsets of attributes, we can generate concepts at different levels of granularity or abstraction. We can arrange these concepts into a *compositional abstraction hierarchy*, as shown in Fig. 6.1. This is a directed acyclic graph (DAG) in which nodes represent concepts, and an edge from a node to its parent is added whenever we drop one of the attributes from the child's concept definition. Note that we dont make any assumptions about the order in which the attributes are dropped (that is, dropping the attribute "smiling" is just as valid as dropping "female" in Fig. 6.1). Thus, the tree shown in the figure is just a subset extracted from the full DAG of concepts, shown for illustration purposes.

One can describe a concept by creating the attribute vector $\mathbf{y}_{\mathcal{O}}$, in which we only

specify the value of the attributes in the subset $\mathcal{O} \subseteq \mathcal{A}$; the remaining attributes are unspecified, and are assumed to take all possible legal values. For example, consider the following concepts, in order of increasing abstraction: $C_{msb} = (\text{male}, \text{smiling}, \text{blackhair})$, $C_{*sb} = (*, \text{smiling}, \text{blackhair})$, and $C_{**b} = (*, *, \text{blackhair})$, where the attributes are gender, smiling or not, and hair color, and $*$ represents "don't care". A good model should be able to generate images from different levels of the abstraction hierarchy, as shown in Fig. 6.1. (This is in contrast to most prior work on conditional generative models of images, which assume that all attributes are fully specified, which corresponds to sampling only from leaf nodes in the hierarchy.)

Ofcourse, this kind of a compositional concept hierarchy is hard to model in general, as understanding the semantics of adjective modifiers like 'laughing' on nouns like 'male' is a tricky phenomenon with rich literature in linguistics (Morzycki 2015). For example, while the set of canadian surgeons is the intersection of the set of all canadians and the set of all surgeons, other concepts do not combine intersectively. Consider the phrase "skillful surgeon", one cannot make the claim then that the phrase is the intersection of all skillful individuals and the set of all surgeons, indeed talking about the set of skillful individuals without first mentioning the skill makes it hard to even define the set in the first place. In this chapter, while I will not make any explicit distinctions, the visual imagination models will generally focus on the class of intersectional interactions between words (Morzycki 2015).

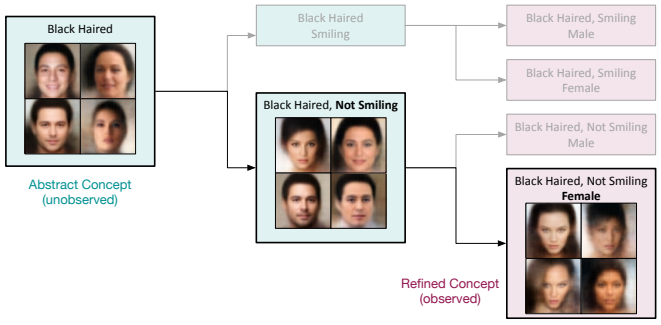


Figure 6.1: A compositional abstraction hierarchy for faces, derived from 3 attributes: hair color, smiling or not, and gender. We show a set of sample images generated by our model, when trained on CelebA, for different nodes in this hierarchy.

In Sec. 6.1, I show how we can extend the variational autoencoder (VAE) framework of Kingma and Welling 2014a to create models which can perform this task. The first extension is to modify the model to the "multi-modal" setting where we have both an image, $\mathbf{x}$, and an attribute vector, $\mathbf{y}$. More precisely, I assume a joint generative model of the form $p(\mathbf{x}, \mathbf{y}, \mathbf{z}) = p(\mathbf{z})p(\mathbf{x}|\mathbf{z})p(\mathbf{y}|\mathbf{z})$, where $p(\mathbf{z})$ is the prior over the latent variable $\mathbf{z}$, $p(\mathbf{x}|\mathbf{z})$ is the image decoder, and $p(\mathbf{y}|\mathbf{z})$ is the description decoder. I additionally assume that the description decoder factorizes over the specified attributes in the description, so $p(\mathbf{y}_{\mathcal{O}}|\mathbf{z}) = \prod_{k \in \mathcal{O}} p(y_k|\mathbf{z})$.

I further extend the VAE by devising a novel objective function, which I call the *TELBO*, for training the model from paired data, $\mathcal{D} = \{(\mathbf{x}_n, \mathbf{y}_n)\}$. However, at test time, the model will need to process unpaired data (either just a description or just an image). Hence I propose to fit three inference networks: $q(\mathbf{z}|\mathbf{x}, \mathbf{y})$, $q(\mathbf{z}|\mathbf{x})$ and $q(\mathbf{z}|\mathbf{y})$. This way we can embed an image or a description into the same shared latent space (using $q(\mathbf{z}|\mathbf{x})$ and $q(\mathbf{z}|\mathbf{y})$, respectively); this lets us "translate" images into descriptions or vice versa, by computing $p(\mathbf{y}|\mathbf{x}) = \int d\mathbf{z}\; p(\mathbf{y}|\mathbf{z})q(\mathbf{z}|\mathbf{x})$ and $p(\mathbf{x}|\mathbf{y}) = \int d\mathbf{z}\; p(\mathbf{x}|\mathbf{z})q(\mathbf{z}|\mathbf{y})$.

To handle abstract concepts (i.e., partially observed attribute vectors), I will use a method based on the product of experts (POE) (Hinton 2002a). In particular, I will use an inference network for attributes has of the form $q(\mathbf{z}|\mathbf{y}_{\mathcal{O}}) \propto p(\mathbf{z}) \prod_{k \in \mathcal{O}} q(\mathbf{z}|\mathbf{y}_k)$. If no attributes are specified, the posterior is equal to the prior. As we condition on more attributes, the posterior becomes narrower, which corresponds to specifying a more precise concept. This will enable us to generate a more diverse set of images to represent abstract concepts, and a less diverse set of images to represent concrete concepts, capturing semantics of intersectional concepts (Morzycki 2015) in a probabilistic latent variable model.

Sec. 6.2 discusses how to evaluate the performance of our method in an objective way. Specifically, I will first "ground" the description by generating a set of images, $\mathcal{S}(\mathbf{y}_{\mathcal{O}}) = \{\mathbf{x}^s \sim p(\mathbf{x}|\mathbf{y}_{\mathcal{O}}) : s = 1 : S\}$. We can then check that all the sampled images in $\mathcal{S}(\mathbf{y}_{\mathcal{O}})$ are consistent with the specified attributes $\mathbf{y}_{\mathcal{O}}$ (I call this **correctness**). We can also

check that the set of images "spans" the extension of the concept, by exhibiting suitable diversity (c.f. Young et al. 2014a). Concretely, we check that the attributes that were *not specified* (e.g., gender in C_{*sb} above) vary across the different images; we call this **coverage**. Finally, we want the set of images to have high correctness and coverage even if the concept $\mathbf{y}_{\mathcal{O}}$ has a combination of attribute values that have not been seen in training. For example, if we train on $C_{msb} = (\text{male}, \text{smiling}, \text{blackhair})$, and $C_{fnb} = (\text{female}, \text{notsmiling}, \text{blackhair})$, we should be able to test on $C_{mnb} = (\text{male}, \text{notsmiling}, \text{blackhair})$, and $C_{fsb} = (\text{female}, \text{smiling}, \text{blackhair})$. I will call this property **compositionality**. Being able to generate plausible images in response to truly compositionally novel queries is the essence of imagination. Together, I will call these criteria *the 3 C's of visual imagination*.

In addition, a model with a good understanding of concepts should be able to identify the concept denoted by a set of images, by naming the concept from the appropriate level in the concept hierarchy (Tenenbaum 1999b). For example, given a set of images of "male with black hair" the model should provide the name "male with black hair" as opposed to the name "person" which is correct, but is not the tightest fit to the concept, or the concept "male with black hair and glasses" which is overly specific and incorrect. This observation motivates a study of concept naming with imagination models in Sec. 6.5.

Sec. 6.5.2 reports experimental results on two different datasets. The first dataset is a modified version of MNIST, which I will call MNIST-with-attributes (or MNIST-A), in which I "render" modified versions of a single MNIST digit on a 64x64 canvas, varying its location, orientation and size. The second dataset is CelebA (Liu et al. 2015), which consists of over 200k face images, annotated with 40 binary attributes. I show that our method outperforms previous methods on these datasets.

In this chapter, I will make the following contributions. First, I will present a novel extension to VAEs in the multimodal setting, introducing a principled new training objective (the TELBO), and deriving an interpretation of a previously proposed objective (JMVAE) (Wang, Lee, and Livescu 2016b) as a valid alternative in Sec. D.1.1. Second, I will present a novel

way to handle missing data in inference networks based on a product of experts. Third, I will present novel criteria (the 3 C's) for evaluating conditional generative models of images, that extends prior work by considering the notion of visual abstraction and imagination, as well as demonstrate applications of imagination models to concept naming.

6.1 Methods

I start by describing standard VAEs, to introduce notation. I will then discuss extensions to handle the multimodal and the missing input settings.

Standard VAEs. A variational autoencoder (Kingma and Welling 2014b) is a latent variable model of the form $p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) = p_{\boldsymbol{\theta}}(\mathbf{z})p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$, where $p_{\boldsymbol{\theta}}(\mathbf{z})$ is the prior (I assume it is Gaussian, $p_{\boldsymbol{\theta}}(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{I})$, although this assumption can be relaxed), and $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ is the likelihood (sometimes called the decoder), usually represented by a neural network. To perform approximate posterior inference, we will fit an inference network (sometimes called the encoder) of the form $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$, so as to maximize $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \mathbb{E}_{\hat{p}(\mathbf{x})}\left[\mathrm{elbo}(\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\phi})\right]$, where $\hat{p}(\mathbf{x}) = \frac{1}{N}\sum_{n=1}^{N} \delta_{\mathbf{x}_n}(\mathbf{x})$ is the empirical distribution, and ELBO is the evidence lower bound:

$$\mathrm{elbo}_{\lambda,\beta}(\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\phi}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x},\boldsymbol{\phi})}\left[\lambda \log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})\right] - \beta \mathrm{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}), p_{\boldsymbol{\theta}}(\mathbf{z})) \tag{6.1}$$

Here $\mathrm{KL}(p, q)$ is the Kullback Leibler divergence between distributions p and q. By default, $\beta = \lambda = 1$, in which case we will just write $\mathrm{elbo}(\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\phi})$. However, by using $\beta > 1$ we can encourage the posterior to be closer to the factorial prior $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{I})$, which encouarges the latent factors to be "disentangled", as proved in (Achille and Soatto 2017); this is known as the β-VAE trick (Higgins et al. 2017b). And allowing $\lambda > 1$ will be useful later, when we have multiple modalities.

Joint VAEs and the TELBO. We will extend the VAE to model images and attributes by defining the joint distribution $p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y}, \mathbf{z}) = p_{\boldsymbol{\theta}}(\mathbf{z})p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{z})$, where $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ is the image decoder (we use the DCGAN architecture from (Radford, Metz, and Chintala 2016)), and $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{z})$ is an MLP for the attribute vector. The corresponding training objective which we want to maximize becomes $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \mathbb{E}_{\hat{p}(\mathbf{x},\mathbf{y})}\left[\text{elbo}(\mathbf{x}, \mathbf{y}, \boldsymbol{\theta}, \boldsymbol{\phi})\right]$, where $\hat{p}(\mathbf{x}, \mathbf{y}) = \frac{1}{N}\sum_{n=1}^{N} \delta_{\mathbf{x}_n}(\mathbf{x})\delta_{\mathbf{y}_n}(\mathbf{y}_n)$ is the empirical distribution derived from paired data, and the joint ELBO is given by

$$\begin{aligned}\text{elbo}_{\lambda_x,\lambda_y,\beta}(\mathbf{x}, \mathbf{y}, \boldsymbol{\theta}_x, \boldsymbol{\theta}_y, \boldsymbol{\phi}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x},\mathbf{y})}\left[\lambda_x \log p_{\boldsymbol{\theta}_x}(\mathbf{x}|\mathbf{z}) + \lambda_y \log p_{\boldsymbol{\theta}_y}(\mathbf{y}|\mathbf{z})\right] \\ - \beta \text{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}, \mathbf{y}), p_{\boldsymbol{\theta}}(\mathbf{z}))\end{aligned}$$

We will call this the JVAE (joint VAE) model. I will usually set $\beta = 1$, but $\lambda_y/\lambda_x > 1$ to to scale up the likelihood from the low dimensional attribute vector, $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{z})$, to match the likelihood from the high dimensional image, $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$.

Having fit the joint model above, we can proceed to train unpaired inference networks $q_{\boldsymbol{\phi}_x}(\mathbf{z}|\mathbf{x})$ and $q_{\boldsymbol{\phi}_y}(\mathbf{z}|\mathbf{y})$, so we can embed images and attributes into the same shared latent space. Keeping the p family fixed from the joint model, a natural objective to fit, say, $q_{\boldsymbol{\phi}_x}(\mathbf{z}|\mathbf{x})$ is to maximize the following:[1]

$$\begin{aligned}\mathcal{L}(\boldsymbol{\phi}_x|\boldsymbol{\theta}) &= -\mathbb{E}_{\hat{p}(\mathbf{x})}\left[\text{KL}(q_{\boldsymbol{\phi}_x}(\mathbf{z}|\mathbf{x}), p_{\boldsymbol{\theta}_x}(\mathbf{z}|\mathbf{x}))\right] \\ &= \int\int d\mathbf{x}d\mathbf{z}\ \hat{p}(\mathbf{x})q_{\boldsymbol{\phi}_x}(\mathbf{z}|\mathbf{x})\left[-\log q_{\boldsymbol{\phi}_x}(\mathbf{z}|\mathbf{x}) - \log p_{\boldsymbol{\theta}_x}(\mathbf{x}) + \log p_{\boldsymbol{\theta}_x}(\mathbf{x}|\mathbf{z}) + \log p_{\boldsymbol{\theta}}(\mathbf{z})\right] \\ &= \mathbb{E}_{\hat{p}(\mathbf{x})}\left[\text{elbo}(\mathbf{x}, \boldsymbol{\theta}_x, \boldsymbol{\phi}_x)\right] - \mathbb{E}_{\hat{p}(\mathbf{x})}\left[\log p_{\boldsymbol{\theta}_x}(\mathbf{x})\right]\end{aligned}$$

where the last term is constant wrt $\boldsymbol{\phi}_x$ and the model family p, and hence can be dropped. We can use a similar method to fit $q_{\boldsymbol{\phi}_y}(\mathbf{z}|\mathbf{y})$. Combining these gives the following triple

[1] A reasonable alternative would be to minimize $\mathbb{E}_{\hat{p}(\mathbf{x})}\left[\text{KL}(p_{\boldsymbol{\theta}_x}(\mathbf{z}|\mathbf{x}), q_{\boldsymbol{\phi}_x}(\mathbf{z}|\mathbf{x}))\right]$. However, this is intractable, since we cannot compute $p_{\boldsymbol{\theta}_x}(\mathbf{z}|\mathbf{x})$, by assumption.

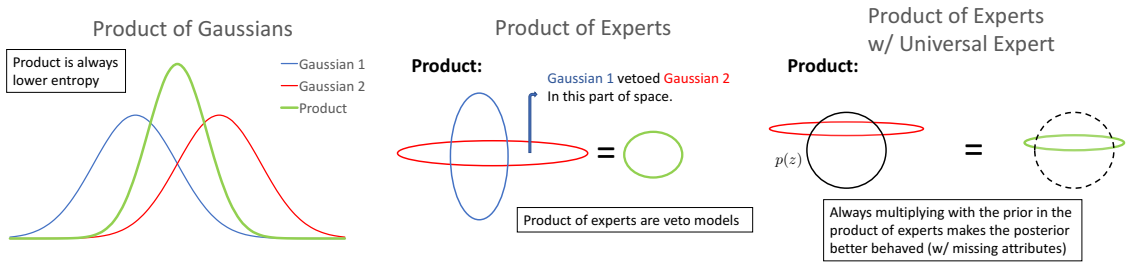


Figure 6.2: Illustration of the product of experts inference network. Each expert votes for a part of latent space implied by its observed attribute. The final posterior is the intersection of these regions. When all attributes are observed, the posterior will be a narrowly defined Gaussian, but when some attributes are missing, the posterior will be broader. Right: we illustrate how inclusion of the "universal expert" $p(\mathbf{z})$ in the product ensures that the posterior is always well-conditioned (close to spherical), even when we are missing some attributes.

ELBO (*TELBO*) objective:

$$\mathcal{L}(\boldsymbol{\theta}_x, \boldsymbol{\theta}_y, \boldsymbol{\phi}, \boldsymbol{\phi}_x, \boldsymbol{\phi}_y) = \mathbf{E}_{\hat{p}(\mathbf{x},\mathbf{y})} \left[\text{elbo}_{1,\lambda,1}(\mathbf{x}, \mathbf{y}, \boldsymbol{\theta}_x, \boldsymbol{\theta}_y, \boldsymbol{\phi}) \right.$$
$$\left. + \text{elbo}_{1,1}(\mathbf{x}, \boldsymbol{\theta}_x, \boldsymbol{\phi}_x) + \text{elbo}_{\gamma,1}(\mathbf{y}, \boldsymbol{\theta}_y, \boldsymbol{\phi}_y)\right] \quad (6.2)$$

where λ and γ scale the log likelihood terms $\log p(\mathbf{y}|\mathbf{z})$; These parameters are set using a validation set. Since we are training the generative model only on aligned data, and simply retrofitting inference networks, I freeze the $p_{\boldsymbol{\theta}_x}(\mathbf{x}|\mathbf{z})$ and $p_{\boldsymbol{\theta}_y}(\mathbf{y}|\mathbf{z})$ terms when training the last two ELBO terms above, and just optimize $q_{\boldsymbol{\phi}_x}(\mathbf{z}|\mathbf{x})$ and $q_{\boldsymbol{\phi}_y}(\mathbf{z}|\mathbf{y})$ terms. This enables optimization of all terms in Eqn. 6.2 jointly. Alternatively, we can first fit the joint model, and then fit the unimodal inference networks.[2] In Sec. 6.3, I compare this to other methods for training joint VAEs that have been proposed in the literature.

Handling missing attributes. In order to handle missing attributes at test time, I use a product of experts model, where each attribute instantiates an expert. This is motivated by prior work (Williams and Nash 2018) which shows that for a linear factor analysis model, the posterior distribution $p(\mathbf{z}|\mathbf{y})$ is a product of K-dimensional Gaussians, one for each

[2] If we have unlabeled image data, we can perform semisupervised learning by optimizing $\mathbb{E}_{\hat{p}(\mathbf{x})} \left[\text{elbo}(\mathbf{x}, \boldsymbol{\theta}_x, \boldsymbol{\phi}_x)\right]$ wrt $\boldsymbol{\theta}_x$ and $\boldsymbol{\phi}_x$, as in (Pu et al. 2016).

visible dimension. Since our model is just a nonlinear extension of factor analysis, I choose the form of the approximate posterior of our inference network, $q(\mathbf{z}|\mathbf{y})$, to be a product of Gaussians, one for each visible feature: $q(\mathbf{z}|\mathbf{y}_{\mathcal{O}}) \propto p(\mathbf{z}) \prod_{k \in \mathcal{O}} q(\mathbf{z}|y_k)$, where $q(\mathbf{z}|y_k) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_k(y_k), \mathbf{C}_k(y_k))$ is the kth Gaussian "expert", and $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_0 = \mathbf{0}, \mathbf{C}_0 = \mathbf{I})$ is the prior. A similar model has been recently proposed in (Bouchacourt, Tomioka, and Nowozin 2018) to perform inference for a set of images. Unlike the product of experts model in (Hinton 2002b), this model multiplies Gaussians, not Bernoullis, so the product has a closed form solution namely $q(\mathbf{z}|\mathbf{y}_{\mathcal{O}}) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}, \mathbf{C})$, where $\mathbf{C}^{-1} = \sum_k \mathbf{C}_k^{-1}$ and $\boldsymbol{\mu} = \mathbf{C}(\sum_k \mathbf{C}_k^{-1} \boldsymbol{\mu}_k)$, and the sum is over all the observed attributes. Intuitively, $\mathbf{y}$ imposes an increasing number of constraints on $\mathbf{z}$ as more of it is observed, as explained in (Williams and Agakov 2002). In our setting, if we do not observe any attributes, the posterior reduces to the prior. As we observe more attributes, the posterior becomes narrower, since the (positive definite) precision matrices, $\mathbf{C}^{-1}$ add up, reflecting the increased specificity of the concept being specified, as illustrated in Fig. 6.2 (middle) (see also (Williams and Agakov 2002)). We will always include the prior term, $p(\mathbf{z})$, in the product, since without it, the posterior $q_{\phi_y}(\mathbf{z}|\mathbf{y}_{\mathcal{O}})$ may not be well-conditioned when we are missing attributes, as illustrated in Fig. 6.2 (right). For more implementation-level details on the model architectures, see Sec. D.1.4.

6.2 Evaluation metrics: The 3C's of Visual Imagination

To evaluate the quality of a set of generated images, $\mathcal{S}(c) = \{\mathbf{x}_s \sim p(\mathbf{x}|c) : s = 1 : S\}$, I apply a multi-label classifier to each image, to convert it to a predicted attribute vector, $\hat{\mathbf{y}}(\mathbf{x})$. This attribute classifier is trained on a large dataset of images and attributes, and is held constant across all methods that are being evaluated. It plays the role of a human observer. This is similar in spirit to generative adversarial networks (Goodfellow et al. 2014b), that declare a generated image to be good enough if a binary classifier cannot distinguish it from a real image. (Both approaches avoid the problems mentioned in (Theis, Oord, and Bethge

2016) related to evaluating generative image models in terms of their likelihood.) However, the attribute classifier checks not only that the images look realistic, but also that they have the desired attributes.

To quantify this, we define the **correctness** as the fraction of attributes for each generated image that match those specified in the concept's description: $\text{correctness}(\mathcal{S}, c) = \frac{1}{|\mathcal{S}|}\sum_{\mathbf{x}\in\mathcal{S}}\frac{1}{|\mathcal{O}|}\sum_{k\in\mathcal{O}}\mathbb{I}(\hat{y}(\mathbf{x})_k = y_k)$. However, we also want to measure the diversity of values for the *unspecified* or missing attributes, $\mathcal{M} = \mathcal{A} \setminus \mathcal{O}$. We can do this by comparing q_k, the empirical distribution over values for attribute k induced by the generated set $\mathcal{S}$, to p_k, the true distribution for this attribute induced by the training set. We will measure the difference between these distributions using the Jensen-Shannon divergence, since it is symmetric and satisfies $0 \leq \text{JS}(p, q) \leq 1$. We can then define the **coverage** as follows: $\text{coverage}(S, c) = \frac{1}{|\mathcal{M}|}\sum_{k\in\mathcal{M}}(1 - \text{JS}(p_k, q_k))$. If desired, we can combine correctness and coverage into a single number, by computing the JS divergence between p_k and q_k for all attributes, where, for observed attributes, p_k is a delta function and q_k is the empirical distribution (I call this **JS-overall**). This gives us a convenient way to pick hyperparameters. However, for analysis, we will report correctness and coverage separately.

Note that this metric is different from the inception score proposed in (Salimans et al. 2016). That is defined as follows: $\text{inception} = \exp\left(\mathbb{E}_{\hat{p}(\mathbf{x})}\left[\text{KL}(p(y|\mathbf{x}), p(y))\right]\right)$, where y is a class label. Expanding the term inside the exponential, we get

$$\sum_{\mathbf{x}} p(\mathbf{x})\left[\sum_{y} p(y|\mathbf{x})\log p(y|\mathbf{x})\right] - \sum_{\mathbf{x}}\sum_{y} p(\mathbf{x}, y)\log p(y) = \mathbb{E}_{\hat{p}(\mathbf{x})}\left[-H(y|\mathbf{x})\right] + H(y)$$

A high inception score means that the distribution $p(y|\mathbf{x})$ has low entropy, so the generated images match some class, but that the marginal $p(y)$ has high entropy, so the images are diverse. However, the inception score was created to evaluate unconditional generative models of images, so it does not check if the generated images are consistent with the concept $\mathbf{y}_\mathcal{O}$, and the degree of diversity does not vary in response to the level of abstraction

of the concept.

Finally, we can assess how well the model understands **compositionality**, by checking correctness of its generated images in response to test concepts $\mathbf{y}_{\mathcal{O}}$ that differ in at least one attribute from the training concepts. I call this a *compositional split* of the data. This is much harder than a standard *iid* split, since we are asking the model to predict the effects of novel combinations of attributes, which it has not seen before (and which might actually be impossible). Note that abstraction is different from compositionality – in abstraction we are asking the model to predict the effects of dropping certain attributes instead of predicting novel combinations of attributes.

6.3 Related Work

In this section, I briefly mention some of the most closely related prior work in the space of generative models and concept learning.

Conditional models. Many conditional generative image models of the form $p(\mathbf{y}|\mathbf{x})$ have been proposed recently, where $\mathbf{y}$ can be a class label (e.g., (Radford, Metz, and Chintala 2016)), a vector of attributes (e.g., (Yan et al. 2016b)), a sentence (e.g., (Reed et al. 2016b)), another image (e.g., (Isola et al. 2017b)), etc. Such models are usually based on VAEs or GANs. However, we are more interested in learning a shared latent space from either descriptions $\mathbf{y}$ or images $\mathbf{x}$, which means we need to use a joint, symmetric, model.

Joint models. Several papers use the same joint VAE model as us, but they differ in how it is trained. In particular, the BiVCCA objective of (Wang, Lee, and Livescu 2016c) has the form $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \mathbb{E}_{\hat{p}(\mathbf{x},\mathbf{y})}\left[J(\mathbf{x}, \mathbf{y}, \boldsymbol{\theta}, \boldsymbol{\phi})\right]$, where

$$
\begin{aligned}
J(\mathbf{x}, \mathbf{y}, \boldsymbol{\theta}, \boldsymbol{\phi}) = \mu \left(E_{q_{\boldsymbol{\phi}_x}(\mathbf{z}|\mathbf{x})}[\log p_{\boldsymbol{\theta}_x}(\mathbf{x}|\mathbf{z}) + \lambda \log p_{\boldsymbol{\theta}_y}(\mathbf{y}|\mathbf{z})] - \mathrm{KL}(q_{\boldsymbol{\phi}_x}(\mathbf{z}|\mathbf{x}), p_{\boldsymbol{\theta}}(\mathbf{z})) \right) \\
+(1-\mu) \left(E_{q_{\boldsymbol{\phi}_y}(\mathbf{z}|\mathbf{y})}[\log p_{\boldsymbol{\theta}_x}(\mathbf{x}|\mathbf{z}) + \lambda \log p_{\boldsymbol{\theta}_y}(\mathbf{y}|\mathbf{z})] - \mathrm{KL}(q_{\boldsymbol{\phi}_y}(\mathbf{z}|\mathbf{y}), p_{\boldsymbol{\theta}}(\mathbf{z})) \right)
\end{aligned}
$$

This method results in the model generating the mean image corresponding to each concept, due to the $E_{q_{\boldsymbol{\phi}_y}(\mathbf{z}|\mathbf{y})} \log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y}|\mathbf{z})$ term, which requires that $\mathbf{z}$'s sampled from $q_{\boldsymbol{\phi}_y}(\mathbf{z}|\mathbf{y}_n)$ be good at generating all the different $\mathbf{x}_n$'s which co-occur with $\mathbf{y}_n$. I show this empirically in Sec. 6.5.2. This problem can be partially compensated for by increasing μ, but that reduces the $\mathrm{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{y}), p_{\boldsymbol{\theta}}(\mathbf{z}))$ penalty, which is required to ensure $q_{\boldsymbol{\phi}_y}(\mathbf{z}|\mathbf{y})$ is a broad distribution with good coverage of the concept.

The JMVAE objective of (Suzuki, Nakayama, and Matsuo 2017b) has the form $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \mathbb{E}_{\hat{p}(\mathbf{x},\mathbf{y})}\left[J(\mathbf{x}, \mathbf{y}, \boldsymbol{\theta}, \boldsymbol{\phi})\right]$, where

$$J(\mathbf{x}, \mathbf{y}, \boldsymbol{\theta}, \boldsymbol{\phi}) = \mathrm{elbo}_{1,\lambda,1}(\mathbf{x}, \mathbf{y}, \boldsymbol{\theta}, \boldsymbol{\phi}) - \alpha\left[\mathrm{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}, \mathbf{y}), q_{\boldsymbol{\phi}_y}(\mathbf{z}|\mathbf{y})) + \mathrm{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}, \mathbf{y}), q_{\boldsymbol{\phi}_x}(\mathbf{z}|\mathbf{x}))\right]$$

At first glance, forcing $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{y})$ to be close to $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}, \mathbf{y})$ seems undesirable, since the latter will typically be close to a delta function, since there is little posterior uncertainty in $\mathbf{z}$ once we see the image $\mathbf{x}$. However, in Sec. D.1.1, I use results from (Hoffman and Johnson 2016) to show that $\mathbb{E}_{\hat{p}(\mathbf{x},\mathbf{y})}\left[\mathrm{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}, \mathbf{y}), q_{\boldsymbol{\phi}_y}(\mathbf{z}|\mathbf{y}))\right]$ can be written in terms of $\mathrm{KL}(q_{\boldsymbol{\phi}}^{\mathrm{avg}}(\mathbf{z}|\mathbf{y}), q_{\boldsymbol{\phi}_y}(\mathbf{z}|\mathbf{y}))$, where $q_{\boldsymbol{\phi}}^{\mathrm{avg}}(\mathbf{z}|\mathbf{y}) = \mathbb{E}_{\hat{p}(\mathbf{x}|\mathbf{y})}\left[q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}, \mathbf{y})\right]$ is the aggregated posterior over $\mathbf{z}$ induced by all images $\mathbf{x}$ which are associated with description $\mathbf{y}$. This ensures that $q_{\boldsymbol{\phi}_y}(\mathbf{z}|\mathbf{y})$ will cover the embeddings of *all* the images associated with concept $\mathbf{y}$. However, since there is no $\mathrm{KL}(q_{\boldsymbol{\phi}_y}(\mathbf{z}|\mathbf{y}), p_{\boldsymbol{\theta}}(\mathbf{z}))$ term, the diversity of the samples is slightly reduced for novel concepts compared to TELBO, shown empirically in Sec. 6.5.2. On the flip side, the benefit of using the aggregated posterior to fit the $q(\mathbf{z}|\mathbf{y})$ inference network is that one can expect sharper images, as this ensures we will sample $\mathbf{z} \sim q(\mathbf{z}|\mathbf{y})$ which have been seen by the image decoder $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ during joint training. If the aggregated posterior does not exactly match the prior (which is known to happen in VAE-type models, see (Hoffman and Johnson 2016)) then regularizing with respect to the prior (as TELBO does) can generate samples in parts of space not seen by the image decoder, which can potentially lead to less "correct" samples. Again, our empirical findings in Sec. 6.5.2 confirm this tradeoff between

correctness and coverage implicit in choices of TELBO *vs.* JMVAE.

The SCAN method of (Higgins et al. 2017c) first fits a standard β-VAE model (Higgins et al. 2017b) on unlabeled images (or rather, features derived from images using a pre-trained denoising autoencoder) by maximizing $\mathcal{L}(\boldsymbol{\theta}_x, \boldsymbol{\phi}_x) = \mathbb{E}_{\hat{p}(\mathbf{x})}\left[\mathrm{elbo}_{1,\beta_x}(\mathbf{x}, \boldsymbol{\theta}_x, \boldsymbol{\phi}_x)\right]$. They then fit a second VAE by maximizing $\mathcal{L}(\boldsymbol{\theta}_y, \boldsymbol{\phi}_y) = \mathbb{E}_{\hat{p}(\mathbf{x},\mathbf{y})}\left[J(\mathbf{x}, \mathbf{y}, \boldsymbol{\theta}_y, \boldsymbol{\phi}_y, \boldsymbol{\phi}_x)\right]$, where

$$J(\mathbf{x}, \mathbf{y}, \boldsymbol{\theta}_y, \boldsymbol{\phi}_y, \boldsymbol{\phi}_x) = \mathrm{elbo}_{1,\beta_y}(\mathbf{y}, \boldsymbol{\theta}_y, \boldsymbol{\phi}_y) - \alpha \mathrm{KL}(q_{\boldsymbol{\phi}_x}(\mathbf{z}|\mathbf{x}), q_{\boldsymbol{\phi}_y}(\mathbf{z}|\mathbf{y}))$$

This is very similar to JMVAE, since $q_{\boldsymbol{\phi}_x}(\mathbf{z}|\mathbf{x}) \approx q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}, \mathbf{y})$, when $(\mathbf{x}, \mathbf{y})$ is a matching pair of images and labels. An important difference, however, is that SCAN treats the attribute vectors $\mathbf{y}$ as atomic symbols; this has the advantage that there is no need to handle missing inputs, but the disadvantage that they cannot infer the meaning of unseen attribute combinations at test time, unless they are "taught" them by having them paired with images. Also, they rely on $\beta_x > 1$ as a way to get compositionality, assuming that a disentangled latent space will suffice. However, in Sec. D.1.3, I show that unsupervised learning of the latent space given images alone can result in poor results when some of the attributes in the compositional concept hierarchy are non-visual, such as parity of an MNIST digit. The proposed approach always takes the labels into consideration when learning the latent space, permitting well-organized latent spaces even in the presence of non-visual concepts (c.f. the difference between PCA and LDA).

Handling missing inputs. Conditional generative models of images, of the form $p(\mathbf{x}|\mathbf{y})$, have problems with missing input attributes, as do inference networks $q(\mathbf{z}|\mathbf{y})$ for VAEs. (Hoffman 2017) uses MCMC to fit a latent Gaussian model, which can in principle handle missing data; however, he initializes the Markov chain with the posterior mode computed by an inference network, which cannot easily handle missing inputs. One approach we can use, if we have a joint model, is to estimate or impute the missing values, as follows: $\hat{\mathbf{y}} = \arg\max_{\mathbf{y}_\mathcal{M}} p(\mathbf{y}_\mathcal{M}|\mathbf{y}_\mathcal{O})$, where $p(\mathbf{y}_\mathcal{M}, \mathbf{y}_\mathcal{O})$ models dependencies between attributes. We

can then sample images using $p(\mathbf{x}|\hat{\mathbf{y}})$. This approach was used in (Yan et al. 2016b) to handle the case where some of the pixels being passed into an inference network were not observed. However, conditioning on an imputed value will give different results from not conditioning on the missing inputs; only the latter will increase the posterior uncertainty in order to correctly represent less precise concepts with broader support.

Gaussian embeddings. There are many papers that embed images and text into points in a vector space. However, we want to represent concepts of different levels of abstraction, and therefore want to map images and text to regions of a (probabilistic) latent space. There are some prior works that use Gaussian embeddings for words (Vilnis and McCallum 2015; Athiwaratkun and Wilson 2017), sometimes in conjunction with images (Mukherjee and Hospedales 2016; Ren et al. 2016). The proposed method differs from these approaches in several ways. First, I maximize the likelihood of $(\mathbf{x}, \mathbf{y})$ pairs, whereas the above methods learn a Gaussian embedding using a contrastive loss. Second, the proposed PoE formulation ensures that the covariance of the posterior $q(\mathbf{z}|\mathbf{y}_{\mathcal{O}})$ is adaptive to the data that is conditioned on. In particular, it becomes narrower as we observe more attributes (because the precision matrices sum up), which is a property not shared by other embedding methods.

Abstraction and compositionality. (Young et al. 2014b) represent the extension of a concept (described by a noun phrase) in terms of a set of images whose captions match the phrase. By contrast, I use a parametric probability distribution in a latent space that can generate new images. (Vendrov et al. 2016) use order embeddings, where they explicitly learn subsumption-like relationships by learning a space that respects a partial order. In contrast, I reason about generality of concepts via the uncertainty induced by their latent representation. There has been some work on compositionality in the language/vision literature (see e.g., (Atzmon et al. 2016; Johnson et al. 2017; Agrawal et al. 2017)), but none of these papers use generative models, which is arguably a much more stringent test of whether a model has truly "understood" the meaning of the components which are being

composed.

Coverage and Submodularity An alternative notion of coverage commonly explored in machine learning is in the context of selecting a subset of items from a universe that maximizes coverage. For example Krause, Singh, and Guestrin 2008, show that one can find constant-factor approximations of optimal sensor placements in a room for maximizing the region of the room where we have satisfactory temperature estimates. Such results often rely on classical work (Nemhauser, Wolsey, and Fisher 1978) which shows that constant-factor approximations are possible using greedy algorithms for a class of functions called submodular functions. Specifically, Krause, Singh, and Guestrin 2008 maximize the mutual information between the temperature at positions in a room, and the placement locations of sensors to derive the optimal set of sensors, for the specific case of gaussian process regression. In the context of this work, one can also view the desiderata of maximizing coverage, as being able to select a set of images $\{\mathbf{x}_i\}$ from the space of all images $\mathcal{X}$, such that the mutual information between the set and the observed variable is maximized, *i.e.* $\max MI(\{\mathbf{x}_i\}, \mathbf{y}_\mathcal{O})$. However, our focus in this work is to evaluate if we learnt a posterior that is appropriately broad for a concept, and not necessarily to draw high-coverage samples/ sets from the learnt posterior. Thus, we evaluate directly for the former and not the latter, computing the coverage metric from Sec. 6.2 on samples drawn IID from the posterior.

6.4 Experimental results

In this section, I will fit the JVAE model to two different datasets (MNIST-A and CelebA), using the TELBO objective, as well as BiVCCA and JMVAE. We will measure the quality of the resulting model using the 3 C's, and show that our method of handling missing data behaves in a qualitatively reasonable way.

6.4.1 MNIST-A

Dataset. In this section, I will report results on the MNIST-A dataset. This is created by modifying the original MNIST dataset as follows. I first create a compositional concept hierarchy using 4 discrete attributes, corresponding to class label (10 values), location (4 values), orientation (3 values), and size (2 values). Thus there are `10x2x3x4=240` unique concepts in total. I then sample ~ 290 example images of each concept, and create both an iid and compositional split of the data. See Sec. D.1.2 for details.

Models and algorithms. I train the JVAE model on this dataset using TELBO, BiVCCA and JMVAE objectives. I use Adam (Kingma and Ba 2015) for optimization, with a learning rate of 0.0001, and a minibatch size of 64. I train all models for 250,000 steps (I generally found that the models do not tend to overfit in the experiments). The models typically take around a day to train on NVIDIA Titan X GPUs. For the image models, $p(\mathbf{x}|\mathbf{z})$ and $q(\mathbf{z}|\mathbf{x})$, I use the DCGAN architecture from (Radford, Metz, and Chintala 2016). The generated images are of size 64×64, as in (Radford, Metz, and Chintala 2016). For the attribute models, $p(y_k|\mathbf{z})$ and $q(\mathbf{z}|y_k)$, I use MLPs. For the joint inference network, $q(\mathbf{z}|\mathbf{x}, \mathbf{y})$, I use a CNN combined with an MLP. I use $d = 10$ latent dimensions for all models. I choose the hyperparameters for each method so as to maximize JS-overall, which is an overall measure of correctness and coverage (see Sec. 6.2) on a validation set of attribute queries. See Sec. D.1.4 for further details on the model architectures.

Evaluation. To measure correctness and coverage, I first train the observation classifier on the full iid dataset, where it gets to an accuracy of 91.18% for class label, 90.56% for scale, 92.23% for orientation, and 100% for location. Consequently, it is a reliable way to assess the quality of samples from various generative models (see Sec. D.1.5 for details). I then compute correctness and coverage on the iid dataset, and coverage on the comp dataset.

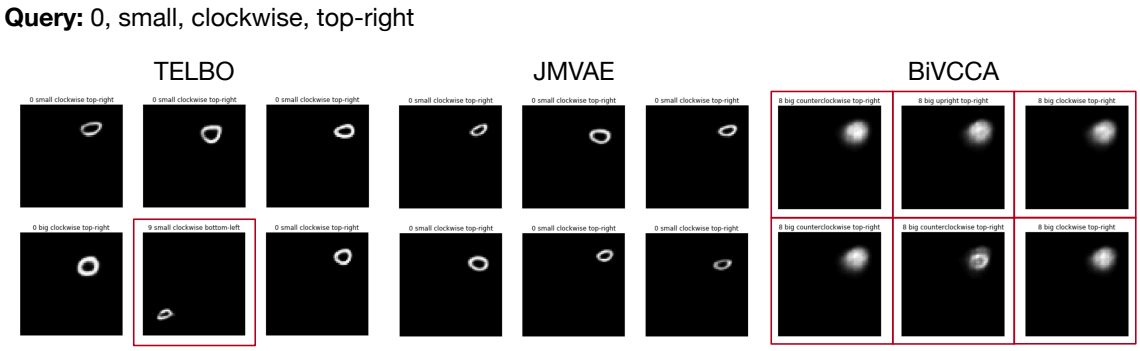


Figure 6.3: Samples from attribute vectors seen at training time, generated by the 3 different models. We plot the posterior mean of each pixel, $\mathbb{E}[\mathbf{x}|\mathbf{z}_s]$, where $\mathbf{z}_s \sim q_{\phi_y}(\mathbf{z}|\mathbf{y})$. The caption at the top of each little image is the predicted attribute values. The border of the generated image is red if any of the attributes are predicted incorrectly. (The observation classifier is fed sampled images, not the mean image that we are showing here.)

Familiar concrete concepts. We will start by assessing the quality of the models in the simplest setting, which is where the test concepts are fully specified (i.e., all attributes are known), and the concepts have been seen before in the training set (i.e., we are using the iid split). Table. 6.1 shows the correctness scores for the three methods. (Since the test concepts are fully grounded, coverage is not well defined, since there are no missing attributes.) We can see that TELBO has a correctness of 82.08%, which is close to that of JMVAE (85.15%); both methods significantly outperform BiVCCA (67.38%). To gain more insight, Fig. 6.3 shows some samples from each of these methods for a leaf concept chosen at random. We can see that the images generated by BiVCCA are very blurry, for reasons we discussed in Sec. 6.3. Note that these blurry images are correctly detected by the attribute classifier.[3] We also see that the JMVAE samples all look good (in this example). Most of the samples from TELBO are also good, although there is one error (correctly detected by the attribute classifier).

Novel abstract concepts. Next we will assess the quality of the models when the test concepts are abstract, i.e., one or more attributes are not specified. (Note that the model was

[3] I chose the value of $\mu = 0.7$ based on maximizing correctness score on the validation set. Nevertheless, this does not completely eliminate blurriness, as we can see.

Table 6.1: I show quantitaive results on the 3C's on MNIST-A. Higher numbers are better. I report standard deviation across 5 splits of the test set.

Method	**#Attributes**	**Coverage** (%)	**Correctness** (%)
		iid	
TELBO	4	-	82.08 $\pm$ 0.56
JMVAE		-	85.15 $\pm$ 0.26
BiVCCA		-	67.38 $\pm$ 0.69
TELBO	3	91.14 $\pm$ 0.53	81.63 $\pm$ 0.38
JMVAE		88.52 $\pm$ 0.37	82.00 $\pm$ 0.37
BiVCCA		85.28 $\pm$ 0.68	70.68 $\pm$ 0.87
TELBO	2	90.32 $\pm$ 0.57	82.03 $\pm$ 1.37
JMVAE		87.89 $\pm$ 0.69	81.02 $\pm$ 1.05
BiVCCA		85.09 $\pm$ 0.76	72.33 $\pm$ 2.31
TELBO	1	90.94 $\pm$ 0.19	83.67 $\pm$ 1.70
JMVAE		88.70 $\pm$ 0.35	81.58 $\pm$ 1.78
BiVCCA		85.53 $\pm$ 0.27	68.36 $\pm$ 2.21
		comp	
TELBO	4	-	75.61 $\pm$ 1.43
JMVAE		-	76.86 $\pm$ 1.30
BiVCCA		-	68.58 $\pm$ 1.02

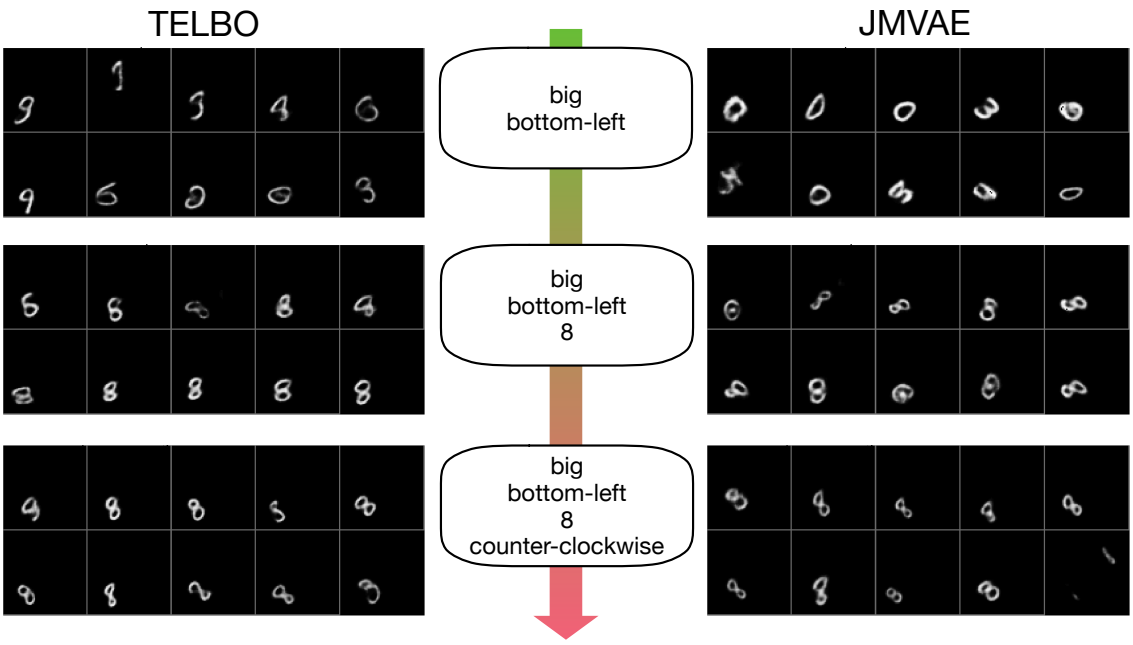


Figure 6.4: Mean images generated by TELBO and JMVAE in response to queries at different levels of abstraction, starting from abstract (top) to refined (bottom), on MNIST-A dataset. For refined/fully specified queries, we can see that both TELBO and JMVAE produce good correctness, *i.e.*, the images produced follow constraints placed by the specified attributes. When the attribute 'orientation' is unspecified, we see that TELBO produces upright and counter clockwise digits, while JMVAE produces clockwise and upright digits. Finally, when the digit is left unspecified (top), we see that TELBO appears to generate a more diverse set of digits (9, 3, 8, 6) while JMVAE produces 0 and 3.

never trained on such abstract concepts.) Table. 6.1 shows that the correctness scores for JMVAE seems to drop somewhat (from about 85% to about 81.5%), although it remains steady for TELBO and BiVCCA. We also see that the coverage of TELBO is higher than the other methods, due to the use of the $\mathrm{KL}(q_{\phi_y}(\mathbf{z}|\mathbf{y}), p_{\boldsymbol{\theta}}(\mathbf{z}))$ regularizer, as discussed in Sec. 6.3. Fig. 6.4 illustrates how the methods respond to concepts of different levels of abstraction. The samples from the TELBO seem to be more diverse, which is consistent with the numbers in Table. 6.1.

Within bin coverage While I focus on diversity along unspecified attributes *i.e.*, on having broader diversity in the digits that are sampled from the model, when the digit is not specified in the query, it is also natural to ask if the model is generating diverse outputs within the bin. Specifically, one can also ask if the model just generates a particular image when

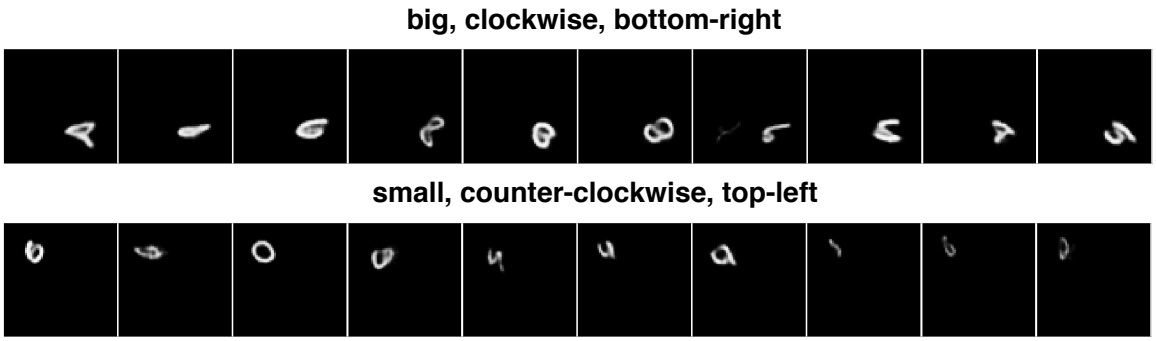


Figure 6.5: An illustration of the diversity of digits generated by the TELBO model when digits are not provided as a label at training time. This illustrates how diverse images produced by the models tend to be in general, without considering any labels into account.

asked to generate the concept $(\text{eight}, \text{small}, \text{upright}, \text{topleft})$, or if it generates a diverse set of images which all satisfy these constraints. Note that measuring such a notion of perceptual diversity in genreal is challenging (Wang et al. 2004); hence I devise a specific solution in context of the current scheme for evaluation, which gives us a sense of the within bin coverage. Specifically, I train the imagination models dropping the digit attribute, training only with the other three attributes in MNIST-A. Thus, the model does not see any annotations informing it of the digit that it is observing at training time. Now, at test time, when asked to generate $(\text{small}, \text{upright}, \text{topleft})$, we can use the observation classifier (which knows how to classify digits) to evaluate if the digits generated tend to be diverse, which gives us a sense of the coverage for this model class *within* the bin. I find that TELBO gets to a coverage of 77.93% and correctness of 87.07%, while JMVAE gets to a coverage of 73.29% and correctness of 87.76% (random chance correctness is 36%). We can observe that the coverage for the "unsupervised" digit factor is lower than the corresponding value when 3 attributes are specified in the supervised case, resulting in a drop from 91.14% (Table. 6.1) to 77.93% coverage in the case of TELBO. However, this is still higher than the worst case coverage one can get, by deterministically picking one of the 10 MNIST digits, which gets to a coverage value of 47.44%. We also find that the digits tend to be fairly diverse from visual inspection. See Fig. 6.5 for more details.

Compositionally novel concrete concepts. Finally I assess the quality of the models when the test concepts are fully specified, but have not been seen before (i.e., we are using the comp split). Table. 6.1 shows some quantitative results. We can see that the correctness for TELBO and JMVAE has dropped from about 82% to about 75%, since this task is much harder, and requires "strong generalization". However, as before, we see that both TELBO and JMVAE outperform BiVCCA, which has a correctness of about 69%. See Sec. D.1.7 qualitative results and more details.

6.4.2 CelebA

In this section, I will report results on the CelebA dataset (Liu et al. 2015). In particular, I use the version that was used in (Perarnau et al. 2016), which selects 18 visually distinctive attributes, and generate images of size 64×64; see Sec. D.1.8 for more details on the CelebA dataset and Sec. D.1.4 for details of the model architectures. Fig. 6.6 shows some sample qualitative results. On the top left, I show some images which were generated by the three methods given the concept shown in the left column. TELBO and JMVAE generate realistic and diverse images. That is, the generated images are generally of males, with mouth slightly open and smiling attributes present in the images. On the other hand, BiVCCA just generates the mean image. On the bottom left, I show what happens when some attributes are dropped, thus specifying more abstract concepts. We can see that when we drop the gender, we get a mixture of both male and female images for both TELBO and JMVAE. Going further, when we drop the "smiling" attribute, we see that the samples now comprise of people who are smiling as well as not smiling, and we see a mixture of genders in the samples. Further, while we see a greater diversity in the samples, we also notice a slight drop in image quality (presumably because none of the approaches has seen supervision with just 'abstract' concepts). See Sec. D.1.9 for more qualitative examples on CelebA.

On the top right, I show some examples of visual imagination, where we ask the models to generate images from the concept "bald female", which does not occur in the training

set.[4] (I omit the results from BiVCCA, which are uniformly poor.) We see that both TELBO and JMVAE can sometimes do a fairly reasonable job (although these are admittedly cherry picked results). Finally, the bottom right illustrates an interesting bias in the dataset: if we ask the model to generate images where we do not specify the value of the eyeglasses attribute, nearly all of the samples fail to included glasses, since the prior probability of this attribute is low (about 6%).

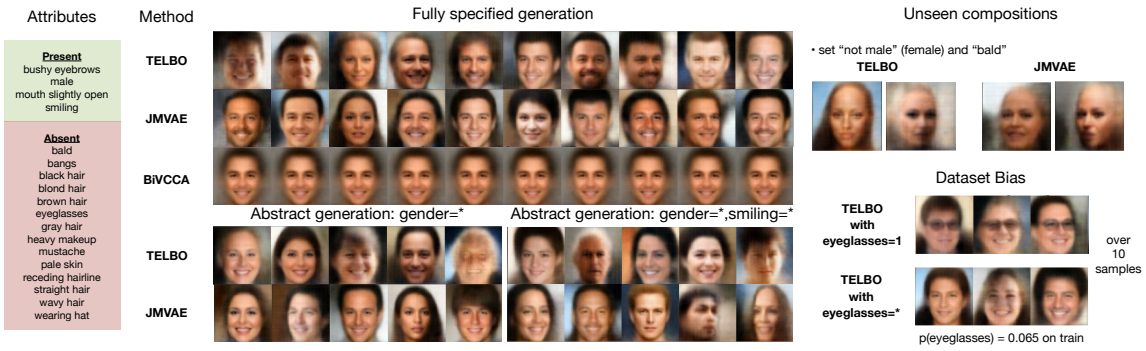


Figure 6.6: Sample CelebA results. Left: I show the attributes specified to be present or absent when generating images. Middle: I show 10 samples each generated from TELBO, JMVAE and BiVCCA. We can see that TELBO and JMVAE genreate better samples than BiVCCA which collapses to the mean. Middle, bottom: We show five samples from TELBO and JMVAE in response to queries with unspecified attributes, and see that both approaches generate a mix in the samples, generalizing meaningfully across unspecified attributes.

6.5 Concept Naming with Imagination Models

In this section, we demonstrate initial results which show that the imagination models can be used for concept naming, where the task is to assign a label to a set of images illustrating the concept depicted by the images. A similar problem has been studied in previous work such as (Tenenbaum 1999a) and (Jia et al. 2013). (Tenenbaum 1999a) studies a set naming problem with integers (instead of images), and show that construct a likelihood function given a hypothesis set that can capture notions of the minimal/smallest hypothesis that explains the observed samples in the set. (Jia et al. 2013) extend this approach to concept-

[4] There are 9 examples in the training set with the attributes (male=0, bald=1), but these turn out to all be labeling errors, as we shown in Sec. D.1.8.

naming on images, incorporating perceptual uncertainty (in recognizing the contents of an image) using a confusion matrix weighted likelihood term. While this approach first extracts labels for each image and then performs concept naming, here we test how well our generative model itself is able to generalize to concept naming without ever performing explicit classification on the images.

In more detail, the problem setup in concept naming is as follows: we are given as input a set $\mathcal{X}$ of images, each of which corresponds to a concept in the compositional abstraction hierarchy 6.1. The task is to assign a label $\mathbf{y} \in \mathcal{Y}$ to the set of images. One of the key challenges in concept learning is to understand "how far" to generalize in the concept hierarchy given a limited number of positive examples (Tenenbaum 1999a). That is, given a small set of images with 7 in the top-left corner and bottom-right corner, one must infer that the concept is "7" as opposed to "7, top-left". In other words, we wish to find the least common ancestor (in the concept hierarchy) corresponding to all the images in the set, given any number of images in the set, so that we can be consistent with the set. We consider two heuristic solutions to this problem:

1. **Concept-NB**: In this approach we compute $\arg\max_{\mathbf{y}} p(\mathbf{y}|\mathcal{X})$, where $p(\mathbf{y}|\mathcal{X})$ is computed using the naive bayes assumption:

$$p(\mathbf{y}|\mathcal{X}) \propto p(\mathbf{y})\Pi_{\mathbf{x}_n \in \mathcal{X}} p(\mathbf{x}_n|\mathbf{y}) = p(\mathbf{y})\Pi_{\mathbf{x}_n \in \mathcal{X}} \int d\mathbf{z}_n p(\mathbf{x}_n|\mathbf{z}_n) q(\mathbf{z}_n|\mathbf{y})$$

   where $p(y)$ is chosen to be uniform across all concepts, and the integrals are approximated using Monte Carlo.

2. **Concept-Latent**: In this approach, instead of working in the observed space, we will work in the latent space. That is, we pick $\arg\min_{\mathbf{y}} \mathrm{KL}(q(\mathbf{z}|\mathcal{X})|q(\mathbf{z}|\mathbf{y}))$, where $q(\mathbf{z}|\mathcal{X})$ is approximated using $\sum_{\mathbf{x}\in\mathcal{X}} q(\mathbf{z}|\mathbf{x})$, which is a mixture of gaussians. The KL divergence can be computed analytically by considering the first two moments of the

gaussian mixture[5].

6.5.1 Experimental Setup

We will use the MNIST-A dataset for the concept naming studies. I consider the fully specified attribute labels in the MNIST-A hierarchy, and consider differrent patterns of missingness (corresponding to different nodes in the abstraction hirearchy) by dropping attributes. Specifically, we will ignore the case where no attribute is specified, and consider a uniform distribution over the rest of the $(2^4 - 1 = 15)$ patterns of missingness. Now, for each fully specified attribute pattern in the iid split of MNIST-A, I sample four missingness patterns and repeat across all fully specified attributes to form a bank of 960 candidate names that a model must choose. I randomly select three subsets of 100 candidate names (and the corresponding images) to form the query set for concept naming, namely tuples of $(\mathbf{y}, \mathcal{X})$. Specifically, given all the images in the eval set for a concept $\mathbf{y}$, we form $\mathcal{X}$ using a randomly sampled subset of 5 images. I will report the accuracy metric, measuring how often the selected concept for a set $\mathcal{X}$ matches the ground truth concept, across three different splits of 100 datapoints.

6.5.2 Results

I evaluate the best versions of TELBO, JMVAE, and BiVCCA on the iid split of MNIST-A for concept naming (table 6.2). In general, Concept-NB approaches perform significantly worse than Concept-Latent approaches. For example, the best Concept-NB approach (using TELBO/BiVCCA objective) gets to an accuracy of around 18%, while Concept-Latent using JMVAE gets to $54.66 \pm 4.92\%$. In general, these numbers are better than a random chance baseline which would get to 0.28% (picking one of 348 effective options, after collating the 960 candidate names based on missingness patterns), while picking the most frequent

[5]Given a Gaussian mixture of the form $g(\mathbf{x}) = \sum_i \pi_i f(\mathbf{x}; \mu_i, \sigma_i)$, where f is the pdf for the Gaussian distribution, the first order moment, that is, the mean of $g(\mathbf{x})$ is given by: $\sum_i \pi_i \mu_i$. The variance is given by: $\sum_i \pi_i \sigma_i^2 + \sum_i \pi_i \mu_i^2 - (\sum_i \pi_i \mu_i)^2$.

Table 6.2: Accuracy of Imagination models on Concept Naming. Higher is better.

Approach	Concept-Latent (%)	Concept-NB (%)
TELBO	35.66 ± 2.05	17.66 ± 1.70
JMVAE	54.66 ± 4.92	13.33 ± 2.05
BiVCCA	28.00 ± 4.54	18.00 ± 1.40
Random	0.28 ± 0.00	0.28 ± 0.00
Most Frequent	6.33 ± 1.88	6.33 ± 1.88

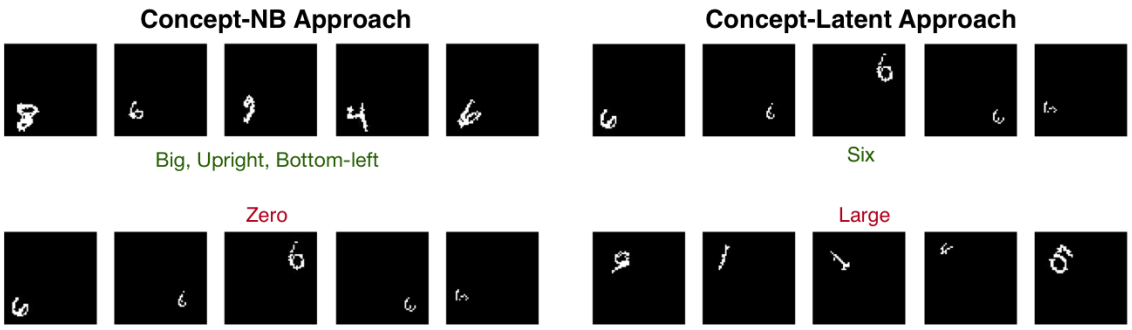


Figure 6.7: A qualitative illustration of some of the examples from concept naming models. Top-left: an example of a sample that is correctly named by a Concept-NB model. However, the Concept-NB model is not that strong and often gets simple concepts such as digits incorrect, making mistakes between 6 and 0, for example (bottom-left). This is likely because the only way in which the Concept-NB approach reasons about the set is not via a "meaningful" low dimensional latent variable but via a sampling distribution on a high dimensional space of images. The Concept-Latent model is able to do better on the same set of images, and classify the set as the concept "6". Finally, I show a failure case of the model where it incorrectly classifies the digits as being large (there is a small digit in the set), and ignores the fact that all of the digits are in the top-left.

(ground truth) fully-specified $\mathbf{y}$ depicted across an image set gets to $6.33 \pm 1.88\%$. fig. 6.7 shows some qualitative examples from Concept-NB as well as Concept-Latent models for concept / set classification. We can observe that the Concept-Latent models are much more powerful than using Concept-NB in terms of naming the concept based on few positive examples from the support set.

CHAPTER 7
CONCLUSION

In this thesis we studied the interactions between vision and language, from the lenses of semantics and pragmatics to support machine learning models which can make human-like inferences. Specifically, we showed that the computational and algorithmic advances from this body of work can enable agents which generate more human-like image captions, which take into account relevant context (pragmatics), better reasoning about commonsense knowledge about our physical world through grounding, and models which can "imagine" abstract attribute vectors (or modifications of concepts) by generating images about those concepts which capture the intension (the definition) and the extension (the span) of the concepts of interest, as well as name the concept denoted by a set of images. Together, these constitute a diverse set of problem domains where we are able to get human-like inferences, with the common thread that often this is possible because we understand something about the computational nature of how to model vision and langauge (or concepts) jointly, or something about the algorithmic setup of how to do joint modeling of vision and language (and often both).

In general, I believe that the research thrust explored in this thesis has value for long-term progress in AI. Given that AI is a hard problem, we need approaches that attack it from various angles, including those that think about conceptual issues in computation, namely what gets computed and what is the input, as well as algorithmic solutions to make current computational solutions more scalable and reliable. I see connecting vision to language and then grounding both of them in actions in an environment as a concrete next step in terms of computational considerations. I think the key algorithmic challenge that will be exciting to see progress on will be a synthesis of generative modeling and reinforcement learning, where agents will choose to do auxilliary generative modeling tasks, and learn to

do inference, and use the same latent state to also perform actions and get rewards. I am excited about the future directions of progress in these areas.

Appendices

APPENDIX A

APPENDIX FOR CIDER: CONSENSUS-BASED IMAGE DESCRIPTION EVALUATION

Appendix Overview

List of items:

1. **Comparison of metrics on triplet annotations to pairwise annotations:** Compares the accuracy of CIDEr on triplet annotation to existing choices of metrics on pairwise annotations

2. **Ranking of reference sentences for various automated metrics:** Qualitative examples of the kind of sentences preferred by each metric

3. **Comparison of rankings from CIDEr and CIDEr-D:** Establishes that both CIDEr and CIDEr-D are similar qualitatively, in terms of how they rank reference sentences

4. **Difference between human-like and what humans like:** Shows examples of differences between pairwise and triplet annotations. Pairwise annotations often favor longer sentences

5. **Sentence collection interface for PASCAL-50S and ABSTRACT-50S:** Shows a snapshot of the interface used to collect our datasets, and explains the instructions

6. **Equations for BLEU, ROUGE, and METEOR:** Formulates some existing metrics in terms of the notation used in the rest of the paper

7. **Qualitative examples of outputs of image description methods evaluated in the paper:** Gives a sense for the kind of outputs produced by each of the image description methods evaluated in the paper

8. **Performance of different versions of metrics on consensus:** Benchmarks the performance of different versions of metrics discussed in the paper at matching human consensus

Appendix 1 : Comparison to Pairwise Annotations

We consider some alternate annotation modalities and compare the performance of present metrics on them with that of CIDEr on consensus. The first such modality is a pairwise interface described as follows. Subjects on Amazon Mechanical Turk (AMT) are shown just the two candidate sentences (B and C) with the image (instead of sentence A), and asked to pick the *better* description out of the two. 11 such human judgments are collected for each such pair. These annotations are collected for the same PASCAL-50S candidate sentences as those used for the triplet experiments in the paper. We compare accuracy on *consensus* for CIDEr to accuracy of other metrics on picking the *better* candidate sentence. We find that ROUGE_L at 5 sentences performs at 75.6% whereas the BLEU_4 version performs at 74.75%. ROUGE_1 and BLEU_1 perform at 73.15% and 73.4% respectively at 5 sentences. With METEOR at 5 sentences, the performance is at 79.5%. In contrast, CIDEr at 48 sentences reaches an accuracy of 84% on consensus. Thus the consensus-based protocol comprising of our proposed metric, dataset and human annotation modality provides more accurate automated evaluation.

Appendix 2 : Ranking of Sentences

We now show a ranking of the 48 sentences collected for a particular image as per the CIDEr, BLEU_1, BLEU_1 without Brevity Penalty and ROUGE_1 scores (Fig. A.1). Each reference sentence is considered in turn as a candidate and scored with the remaining (47) reference sentences using the corresponding metric. Note how the top-ranked CIDEr sentences show high consensus. The top-ranked ROUGE sentences are typically more detailed, whereas the top ranked BLEU sentences are not as consistent as those with CIDEr.

If BLEU was used without the brevity penalty, as some previous works have (Kulkarni et al. 2011; Ordonez, Kulkarni, and Berg 2011) one would see that really short sentences get high scores. Intuitively, we can see that the ranking produced by CIDEr is more meaningful.

Appendix 3 : Difference between Human-like and What Humans Like

In our experiments, we found that there can often be a difference in the sentence that is rated as "better" (measured via pairwise annotation) by subjects *versus* the kind of sentences written by subjects when asked to describe the image (measured via consensus annotation). We refer to this distinction as human-like vs what humans like. Some qualitative examples are shown in Fig. A.3. Candidate sentences shown in bold are those that the consensus-based measure picks and those shown in thin font are those picked by the pairwise evaluation based on "better". Reference sentences rated similar to the winning candidate sentence using the triplet annotation are shown in bold.

Appendix 4 : Ranking of sentences - CIDEr and CIDEr-D

As we report in Sec. 4.1.6, we find that CIDEr and CIDEr-D agree with a high correlation (Spearman's ρ=0.94) on ranking of sentences. We now compare CIDEr $_1$ and CIDEr-D$_1$ rankings, since results are easier to interpret for the unigram case. An example of ranking can be found in Fig. A.2. Notice that the rankings of CIDEr and CIDEr-D are very similar qualitatively. However, the formulation of CIDEr-D avoids gaming effects as explained in Sec. 4.1.6.

Appendix 5 : Sentence Collection Interface

The sentence collection interface for both ABSTRACT-50S and PASCAL-50S is shown in Fig. A.4. Stringent rejection criteria were specified.

Appendix 6 : Image Description Method Outputs

In the paper, we compared the relative performance of five image description methods: Midge (Mitchell, Han, and Hayes 2012), Babytalk (Kulkarni et al. 2011), Story (Farhadi et al. 2010), and two versions of Translating Video Content to Natural Language Descriptions (Rohrbach et al. 2013) (Video and Video+). Here, we show a sample image with the descriptions generated by the five methods compared in the paper (Fig. A.5). We can see that Midge (Mitchell, Han, and Hayes 2012) and Babytalk (Kulkarni et al. 2011) produce the better descriptions on this image, consistent with our finding in the paper.

Appendix 7 : Other Metrics

Our goal is to automatically evaluate for an image I_i how well a candidate sentence c_i matches the consensus of a set of image descriptions $S_i = \{s_{i1}, \ldots, s_{im}\}$. The sentences are represented using sets of n-grams, where an n-gram $\omega_k \in \Omega$ is a set of one or more ordered words. In this paper we explore n-grams with one to four words. Each word in an n-gram is modified to its stemming or root form. That is, "fishes", "fishing " and "fished" all get reduced to "fish". The number of times an n-gram ω_k occurs in a sentence s_{ij} is denoted $h_k(s_{ij})$ or $h_k(c_i)$ for the candidate sentence $c_i \in C$.

BLEU

BLEU (Papineni et al. 2002) is a popular machine translation metric that analyzes the co-occurrences of n-grams between the candidate and reference sentences. We compute the sentence level BLEU scores between a candidate sentence and a set of reference sentences. The BLEU score is computed as follows:

$$P_n(c_i, S_i) = \frac{\sum_k \min(h_k(c_i), \max_{j \in m} h_k(s_{ij}))}{\sum_k h_k(c_i)}, \tag{A.1}$$

where k indexes the set of possible n-grams of length n. The clipped precision metric limits the number of times an n-gram may be counted to the maximum number of times it is observed in a single reference sentence. Note that P_n is a precision score and it favors short sentences. So a brevity penalty is also used:

$$b(C,S) = \begin{cases} 1 & \text{if } l_C > l_S \\ e^{1-l_S/l_C} & \text{if } l_C \leq l_S \end{cases}, \tag{A.2}$$

where l_C is the total length of candidate sentences c_i's and l_S is the length of the corpus-level effective reference length. When there are multiple references for a candidate sentence, we choose to use the *closest* reference length for the brevity penalty.

The overall BLEU score is computed using a weighted geometric mean of the individual n-gram precision:

$$BLEU_N(c_i, S_i) = b(c_i, S_i) \exp\left(\sum_{n=1}^{N} w_n \log P_n(c_i, S_i)\right), \tag{A.3}$$

where $N = 1, 2, 3, 4$ and w_n is typically held constant for all n.

BLEU has shown good performance for corpus-level comparisons over which a high number of n-gram matches exist. However, at a sentence-level the n-gram matches for higher n rarely occur. As a result, BLEU performs poorly when comparing individual sentences.

ROUGE

ROUGE is a set of evaluation metrics designed to evaluate text summarization algorithms.

1. ROUGE$_N$: The first ROUGE metric computes a simple n-gram recall over all reference summaries given a candidate sentence:

$$ROUGE_N(c_i, S_i) = \frac{\sum_j \sum_k \min(h_k(c_i), h_k(s_{ij}))}{\sum_j \sum_k h_k(s_{ij})} \tag{A.4}$$

2. ROUGE$_L$: ROUGE$_L$ uses a measure based on the Longest Common Subsequence (LCS). An LCS is a set words shared by two sentences which occur in the same order. However, unlike n-grams there may be words in between the words that create the LCS. Given the length $l(c_i, s_{ij})$ of the LCS between a pair of sentences, ROUGE$_L$ is found by computing an F-measure:

$$R_l = \max_j \frac{l(c_i, s_{ij})}{|s_{ij}|} \tag{A.5}$$

$$P_l = \max_j \frac{l(c_i, s_{ij})}{|c_i|} \tag{A.6}$$

$$ROUGE_L(c_i, S_i) = \frac{(1+\beta^2)R_l P_l}{R_l + \beta^2 P_l} \tag{A.7}$$

R_l and P_l are recall and precision of LCS. β is usually set to favor *recall* ($\beta = 2$). Since n-grams are implicit in this measure due to the use of the LCS, they need not be specified.

3. ROUGE$_S$: The final ROUGE metric uses skip bi-grams instead of the LCS or n-grams. Skip bi-grams are pairs of ordered words in a sentence. However, similar to the LCS, words may be skipped between pairs of words. Thus, a sentence with 4 words would have $C_2^4 = 6$ skip bi-grams. Precision and recall are again incorporated to compute an F-measure score. If $f_k(s_{ij})$ is the skip bi-gram count for sentence s_{ij}, ROUGE$_S$ is computed as:

$$R_s = \max_j \frac{\sum_k \min(f_k(c_i), f_k(s_{ij}))}{\sum_k f_k(s_{ij})} \tag{A.8}$$

$$P_s = \max_j \frac{\sum_k \min(f_k(c_i), f_k(s_{ij}))}{\sum_k f_k(c_i)} \tag{A.9}$$

$$ROUGE_S(c_i, S_i) = \frac{(1+\beta^2)R_s P_s}{R_s + \beta^2 P_s} \tag{A.10}$$

Skip bi-grams are capable of capturing long range sentence structure. In practice, skip bi-grams are computed so that the component words occur at a distance of at most 4 from each other.

METEOR

METEOR is calculated by generating an alignment between the words in the candidate and reference sentences, with an aim of 1:1 correspondence. This alignment is computed while minimizing the number of chunks, ch, of contiguous and identically ordered tokens in the sentence pair. The alignment is based on exact token matching, followed by WordNet synonyms and then stemmed tokens. Given a set of alignments, m, the METEOR score is the harmonic mean of precision and recall between the best scoring reference and candidate:

$$Pen = \gamma \left(\frac{ch}{m} \right)^{\theta} \tag{A.11}$$

$$F_{mean} = \frac{P_m R_m}{\alpha P_m + (1-\alpha) R_m} \tag{A.12}$$

$$P_m = \frac{|m|}{\sum_k h_k(c_i)} \tag{A.13}$$

$$R_m = \frac{|m|}{\sum_k h_k(s_{ij})} \tag{A.14}$$

$$METEOR = (1 - Pen) F_{mean} \tag{A.15}$$

Thus, the final METEOR score includes a penalty based on chunkiness of resolved matches and a harmonic mean term that gives the quality of the resolved matches.

Appendix 8 : Detailed Evaluation

We now show the results for different versions of each metric in the family of BLEU and ROUGE metrics, along with some variations of CIDEr. We use only one (latest) version of METEOR, thus it is not a part of this evaluation. The versions of CIDEr shown here are as follows. **CIDEr exp** refers to an exponential combination of scores obtained by varying

n-gram counts w_n instead of taking a mean, which we describe in Sec. 4.1.2. **CIDEr max** refers to taking a max across scores with different reference sentences, instead of the mean we discuss in the paper. **CIDEr no idf** version sets uniform IDF weights in CIDEr. The rest of the versions of other metrics are explained in the previous section. The results on PASCAL-50S are shown in Fig. A.6 and ABSTRACT-50S are shown in Fig. A.7. We find that removing the IDF weights in the **CIDEr no idf** version hurts performance significantly. **CIDEr max** and **CIDEr exp** perform slightly worse than CIDEr. The best performing version of each of these metrics was discussed in Sec. 5.1.5.

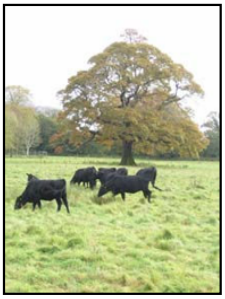
CIDEr	ROUGE	BLEU w/o BP	BLEU
[1] A man is fishing in a canoe on a lake. **[2]** A man fishing in a canoe on a lake **[3]** A man in canoe fishing on a lake **[4]** A man in his canoe fishing on the lake. **[5]** A man fishes in a canoe in an empty lake **[6]** A man fishing off of his canoe on a lake. **[7]** A person in a canoe fishes on a lake. **[8]** a person fishes while sitting in a canoe on a lake **[9]** The man is fishing on a canoe **[10]** A man in a canoe is fishing. **[11]** A man fishing in a canoe. **[12]** man fishing in a canoe **[13]** Someone is fishing from a canoe on a lake. **[14]** a man fishing out of a canoe **[15]** A man in a canoe is fishing on a still lake. **[16]** A man is fishing on a lake. **[17]** A person is fishing from a canoe. **[18]** A man is fishing in a boat in lake **[19]** A man is on a boat fishing on the lake. **[20]** A man is fishing in a small boat on a lake. **[21]** One man fishes in a small boat on the lake. **[22]** A man is fishing from a boat on a lake. **[23]** A man in a canoe fishing in a calm lake. **[24]** A man is fishing alone in a canoe. **[25]** a man fishing in the middle of a lake in a boat **[26]** A person in a canoe fishing on a lake surrounded by hills. **[27]** a person fishing on a lake **[28]** A person is fishing in a boat on a lake. **[29]** Man in a boat fishing. **[30]** A man fishing alone on the lake. **[31]** A man fishes from his small boat. **[32]** A man is fishing from his canoe on quiet water. **[33]** A man is fishing in the river. **[34]** A man on a canoe fishing near a landmass. **[35]** A man is fishing alone on a small boat. **[36]** A lone man sits in a boat and fishes. **[37]** A guy is canoeing and fishing the middle of a tranquil and calm lake. **[38]** A man is out fishing from a canoe on a tranquil morning. **[39]** A man fishing in a kayak. **[40]** A man is fishing in the sea by a forest. **[41]** There is a man in the canoe. **[42]** A person is fishing in the water all by themselves. **[43]** A person is sitting in a boat on a lake. **[44]** A small boat in the middle of the lake. **[45]** A lone fisherman sits in his canoe on a river. **[46]** A lone fisherman sits in a canoe with a pole in the water. **[47]** A man is rowing a man in a river. **[48]** A lone fisherman in a rowboat on an empty lake.	**[1]** A man is fishing in a canoe on a lake. **[2]** A man in a canoe is fishing on a still lake. **[3]** A man is fishing in a small boat on a lake. **[4]** A man fishing in a canoe on a lake **[5]** A man in canoe fishing on a lake **[6]** a man fishing in the middle of a lake in a boat **[7]** A person is fishing in a boat on a lake. **[8]** A man is fishing from a boat on a lake. **[9]** A man in a canoe fishing in a calm lake. **[10]** a person fishes while sitting in a canoe on a lake **[11]** A person in a canoe fishing on a lake surrounded by hills. **[12]** A man is on a boat fishing on the lake. **[13]** A person in a canoe fishes on a lake. **[14]** A man is fishing in a boat in lake **[15]** A man is out fishing from a canoe on a tranquil morning. **[16]** A man fishes in a canoe in an empty lake **[17]** A man in his canoe fishing on the lake. **[18]** A man fishing off of his canoe on a lake. **[19]** A man is fishing alone in a canoe. **[20]** A man in a canoe is fishing. **[21]** One man fishes in a small boat on the lake. **[22]** A man is fishing on a lake. **[23]** Someone is fishing from a canoe on a lake. **[24]** A person is sitting in a boat on a lake. **[25]** A man is fishing in the sea by a forest. **[26]** A man is fishing alone on a small boat. **[27]** A man on a canoe fishing near a landmass. **[28]** A guy is canoeing and fishing the middle of a tranquil and calm lake. **[29]** A man fishing in a canoe. **[30]** A lone man sits in a boat and fishes. **[31]** The man is fishing on a canoe **[32]** A man is fishing from his canoe on quiet water. **[33]** A man is fishing in the river. **[34]** A man fishing in a kayak. **[35]** a man fishing out of a canoe **[36]** A man is rowing a man in a river. **[37]** A man fishing alone on the lake. **[38]** A person is fishing from a canoe. **[39]** A lone fisherman sits in a canoe with a pole in the water. **[40]** a person fishing on a lake **[41]** A lone fisherman sits in his canoe on a river. **[42]** man fishing in a canoe **[43]** A lone fisherman in a rowboat on an empty lake. **[44]** There is a man in the canoe. **[45]** Man in a boat fishing. **[46]** A person is fishing in the water all by themselves. **[47]** A man fishes from his small boat. **[48]** A small boat in the middle of the lake.	**[1]** A man is fishing on a lake. **[2]** A man fishing in a canoe. **[3]** man fishing in a canoe **[4]** A man in a canoe is fishing. **[5]** The man is fishing on a canoe **[6]** A man fishing in a canoe on a lake **[7]** A man in canoe fishing on a lake **[8]** A man is fishing in a canoe on a lake. **[9]** Man in a boat fishing. **[10]** a man fishing out of a canoe **[11]** a person fishing on a lake **[12]** A man in his canoe fishing on the lake. **[13]** A man is fishing in a boat in lake **[14]** A man is on a boat fishing on the lake. **[15]** A person in a canoe fishes on a lake. **[16]** A man is fishing in a small boat on a lake. **[17]** One man fishes in a small boat on the lake. **[18]** A man fishes in a canoe in an empty lake **[19]** A man is fishing from a boat on a lake. **[20]** A man is fishing in the river. **[21]** A person is fishing from a canoe. **[22]** A man fishing off of his canoe on a lake. **[23]** A man is fishing alone in a canoe. **[24]** A man fishing alone on the lake. **[25]** A person is fishing in a boat on a lake. **[26]** A man in a canoe is fishing on a still lake. **[27]** a person fishes while sitting in a canoe on a lake **[28]** a man fishing in the middle of a lake in a boat **[29]** Someone is fishing from a canoe on a lake. **[30]** There is a man in the canoe. **[31]** A man in a canoe fishing in a calm lake. **[32]** A man fishes from his small boat. **[33]** A man fishing in a kayak. **[34]** A lone man sits in a boat and fishes. **[35]** A man is fishing alone on a small boat. **[36]** A man on a canoe fishing near a landmass. **[37]** A person in a canoe fishing on a lake surrounded by hills. **[38]** A man is fishing from his canoe on quiet water. **[39]** A man is fishing in the sea by a forest. **[40]** A person is sitting in a boat on a lake. **[41]** A man is out fishing from a canoe on a tranquil morning. **[42]** A guy is canoeing and fishing the middle of a tranquil and calm lake. **[43]** A small boat in the middle of the lake. **[44]** A person is fishing in the water all by themselves. **[45]** A lone fisherman sits in his canoe on a river. **[46]** A lone fisherman sits in a canoe with a pole in the water. **[47]** A man is rowing a man in a river. **[48]** A lone fisherman in a rowboat on an empty lake.	**[1]** Man in a boat fishing. **[2]** a man fishing out of a canoe **[3]** A person is fishing in a boat on a lake. **[4]** A man is fishing in the river. **[5]** A man fishing in a canoe on a lake **[6]** A man fishes in a canoe in an empty lake **[7]** A man in a canoe is fishing. **[8]** A person in a canoe fishes on a lake. **[9]** A man in his canoe fishing on the lake. **[10]** a man fishing in the middle of a lake in a boat **[11]** A man in canoe fishing on a lake **[12]** A man is fishing in a small boat on a lake. **[13]** A man fishing alone on the lake. **[14]** A man is fishing on a lake. **[15]** A man is fishing from a boat on a lake. **[16]** a person fishing on a lake **[17]** A man is fishing alone on a small boat. **[18]** A person is sitting in a boat on a lake. **[19]** A man fishes from his small boat. **[20]** A man is fishing in a boat in lake **[21]** A man is fishing alone in a canoe. **[22]** A man is fishing in a canoe on a lake. **[23]** The man is fishing on a canoe **[24]** A man is on a boat fishing on the lake. **[25]** One man fishes in a small boat on the lake. **[26]** A man in a canoe fishing in a calm lake. **[27]** A lone man sits in a boat and fishes. **[28]** A man fishing in a canoe. **[29]** A lone fisherman sits in his canoe on a river. **[30]** A person is fishing from a canoe. **[31]** man fishing in a canoe **[32]** A man is out fishing from a canoe on a tranquil morning. **[33]** a person fishes while sitting in a canoe on a lake **[34]** A man in a canoe is fishing on a still lake. **[35]** A lone fisherman in a rowboat on an empty lake. **[36]** A man is fishing from his canoe on quiet water. **[37]** A man fishing off of his canoe on a lake. **[38]** Someone is fishing from a canoe on a lake. **[39]** A small boat in the middle of the lake. **[40]** A guy is canoeing and fishing the middle of a tranquil and calm lake. **[41]** There is a man in the canoe. **[42]** A lone fisherman sits in a canoe with a pole in the water. **[43]** A person in a canoe fishing on a lake surrounded by hills. **[44]** A man fishing in a kayak. **[45]** A man is fishing in the sea by a forest. **[46]** A person is fishing in the water all by themselves. **[47]** A man is rowing a man in a river. **[48]** A man on a canoe fishing near a landmass.

Figure A.1: Ranking of 48 sentences, from highest score to lowest score, as predicted by each metric. Notice how CIDEr captures how most humans tend to describe an image (consensus) better, wheareas ROUGE scores invariably favor longer, detailed sentences (less salient) and BLEU scores favor shorter sentences (lacking coverage) when used without Brevity Penalty. ROUGE_1 and BLEU_1 versions of ROUGE and BLEU are used.

CIDEr	CIDEr-D
[1] A man is fishing in a canoe on a lake. **[2]** A man fishing in a canoe on a lake **[3]** A man in canoe fishing on a lake **[4]** A man in his canoe fishing on the lake. **[5]** A man fishes in a canoe in an empty lake **[6]** A man fishing off of his canoe on a lake. **[7]** A person in a canoe fishes on a lake. **[8]** a person fishes while sitting in a canoe on a lake **[9]** The man is fishing on a canoe **[10]** A man in a canoe is fishing. **[11]** A man fishing in a canoe. **[12]** man fishing in a canoe **[13]** Someone is fishing from a canoe on a lake. **[14]** a man fishing out of a canoe **[15]** A man in a canoe is fishing on a still lake. **[16]** A man is fishing on a lake. **[17]** A person is fishing from a canoe. **[18]** A man is fishing in a boat in lake **[19]** A man is on a boat fishing on the lake. **[20]** A man is fishing in a small boat on a lake. **[21]** One man fishes in a small boat on the lake. **[22]** A man is fishing from a boat on a lake. **[23]** A man in a canoe fishing in a calm lake. **[24]** A man is fishing alone in a canoe. **[25]** a man fishing in the middle of a lake in a boat **[26]** A person in a canoe fishing on a lake surrounded by hills. **[27]** a person fishing on a lake **[28]** A person is fishing in a boat on a lake. **[29]** Man in a boat fishing. **[30]** A man fishing alone on the lake. **[31]** A man fishes from his small boat. **[32]** A man is fishing from his canoe on quiet water. **[33]** A man is fishing in the river. **[34]** A man on a canoe fishing near a landmass. **[35]** A man is fishing alone on a small boat. **[36]** A lone man sits in a boat and fishes. **[37]** A guy is canoeing and fishing the middle of a tranquil and calm lake. **[38]** A man is out fishing from a canoe on a tranquil morning. **[39]** A man fishing in a kayak. **[40]** A man is fishing in the sea by a forest. **[41]** There is a man in the canoe. **[42]** A person is fishing in the water all by themselves. **[43]** A person is sitting in a boat on a lake. **[44]** A small boat in the middle of the lake. **[45]** A lone fisherman sits in his canoe on a river. **[46]** A lone fisherman sits in a canoe with a pole in the water. **[47]** A man is rowing a man in a river. **[48]** A lone fisherman in a rowboat on an empty lake.	**[1]** A man fishing in a canoe on a lake **[2]** A man in canoe fishing on a lake **[3]** A man in his canoe fishing on the lake. **[4]** A man is fishing in a canoe on a lake. **[5]** A person in a canoe fishes on a lake. **[6]** A man fishes in a canoe in an empty lake **[7]** A man fishing off of his canoe on a lake. **[8]** Someone is fishing from a canoe on a lake. **[9]** The man is fishing on a canoe **[10]** A man in a canoe is fishing. **[11]** a person fishes while sitting in a canoe on a lake **[12]** a man fishing out of a canoe **[13]** A man is fishing in a boat in lake **[14]** A man is on a boat fishing on the lake. **[15]** One man fishes in a small boat on the lake. **[16]** A man is fishing from a boat on a lake. **[17]** A man in a canoe is fishing on a still lake. **[18]** A man is fishing on a lake. **[19]** A man fishing in a canoe. **[20]** A man is fishing alone in a canoe. **[21]** A man in a canoe fishing in a calm lake. **[22]** A person is fishing from a canoe. **[23]** A man is fishing in a small boat on a lake. **[24]** A person is fishing in a boat on a lake. **[25]** man fishing in a canoe **[26]** a man fishing in the middle of a lake in a boat **[27]** A man fishing alone on the lake. **[28]** a person fishing on a lake **[29]** A person in a canoe fishing on a lake surrounded by hills. **[30]** A man fishes from his small boat. **[31]** A man is fishing from his canoe on quiet water. **[32]** A man is fishing alone on a small boat. **[33]** A man on a canoe fishing near a landmass. **[34]** A lone man sits in a boat and fishes. **[35]** A man is fishing in the river. **[36]** Man in a boat fishing. **[37]** A man is fishing in the sea by a forest. **[38]** A man is out fishing from a canoe on a tranquil morning. **[39]** A man fishing in a kayak. **[40]** There is a man in the canoe. **[41]** A guy is canoeing and fishing the middle of a tranquil and calm lake. **[42]** A person is fishing in the water all by themselves. **[43]** A person is sitting in a boat on a lake. **[44]** A small boat in the middle of the lake. **[45]** A lone fisherman sits in his canoe on a river. **[46]** A lone fisherman sits in a canoe with a pole in the water. **[47]** A lone fisherman in a rowboat on an empty lake. **[48]** A man is rowing a man in a river.

Figure A.2: Ranking of 48 sentences, from highest score to lowest score, as predicted by CIDEr_1 and CIDEr-D_1. Notice that the rankings are mostly similar qualitatively. CIDEr-D is more robust to gaming effects than CIDEr.

Reference Sentences	Candidate Sentences
A baby girl laughs at the camera **A woman is getting a baby girl to smile for the camera.** **A mom is smiling with a baby.** **A woman sits down next to a baby sitting on the table.** **A woman smiles at a baby who is sitting on a table.** **A woman sits with a baby at a table.** **A baby girl is sitting on a table and smiling.** **A baby is sitting on the counter smiling while her mom looks on.** A woman in spongebob scrub is smiling at a baby in a blue dress. A baby is sitting on a table with her blond mom smiling at her.	**[1] A woman with a smiling baby sitting on the table.** **[2]** A tiny blond child in a blue dress sits on a table near her mother.



Multiple cows graze in the open field of grass. **Black cows graze in the pasture.** **Black cows graze in a green pasture.** **Cows are grazing in a grassy field.** **Black cows are eating a lot of grass.** **A herd of cows eats grass.** **Black cows are grazing in a field.** **Several black cows wander in a green pasture.** Cattle graze in a green pasture near a tall tree. Black cows are grazing in a field in front of a tree.	**[1]** A number of black cows grazing in front of a large tree. **[2] Black cows graze on green grass.**

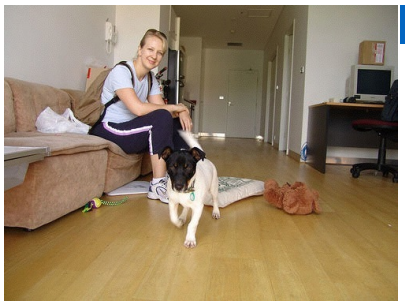
A dog sitting idly on a floral pattern chair. **A little dog sits on a flower cushion.** **A dog relax on a flower patterned chair outside.** **A dog with bell collar sits on a flower pillow.** **A dog lying on a flower patterned chair.** **A dog sitting on a floral chair.** **A brown and white dog is lying on a floral print chair.** **A dog is lying on a flower couch.** **A small dog lying on a flowery cushion stares at the camera.** A dog with a bell collar sits on the chair	**[1]** Brown and white dog with a bell on black collar. **[2] A small orange and white dog with a collar and a bell relaxing on a flower print pillow.**

Figure A.3: Reference sentences shown in **bold** are those which are rated as more similar to the winning candidate sentence, also shown in **bold**, via the triplet interface. The candidate sentence not shown in bold is the one picked by the pairwise interface, which captures "better". This illustrates the difference between human-like *versus* what humans like.

Please describe what is going on in this image.

Sentence:

Figure A.4: Interface used for collecting image descriptions



Midge	Babytalk	Story	Video	Video+
a person with the dog with the sofa	This is a picture of one person, one sofa and one dog. The person is against the brown sofa. The dog is near the person, and beside the brown sofa.	China doll in a leather recliner.	people posing in a restaurant	a man at a table at a restaurant

Figure A.5: Descriptions produced by Midge (Mitchell, Han, and Hayes 2012), Babytalk (Kulkarni et al. 2011), Story (Farhadi et al. 2010), Video (Rohrbach et al. 2013) and Video+ (Rohrbach et al. 2013) for an image. Note that since Story is a retrieval based approach, we consider the top-ranked output to show here.

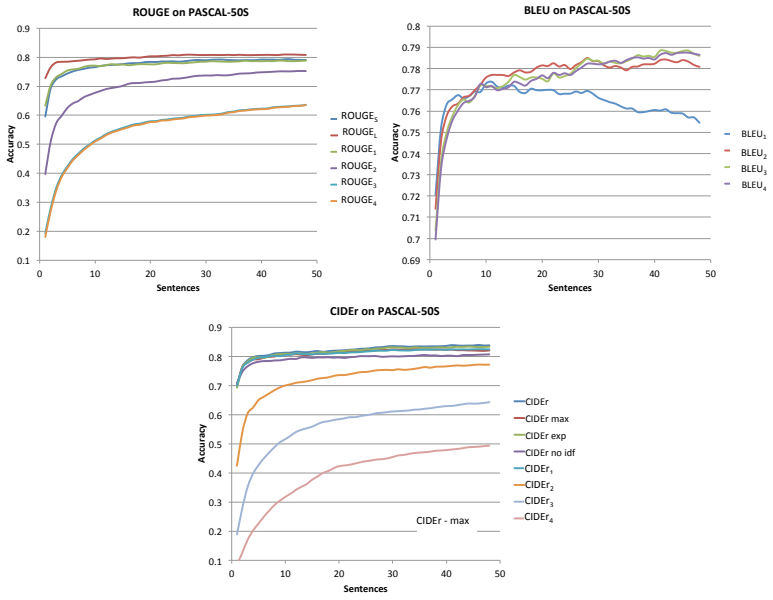


Figure A.6: Performance of different versions of metrics on PASCAL-50S

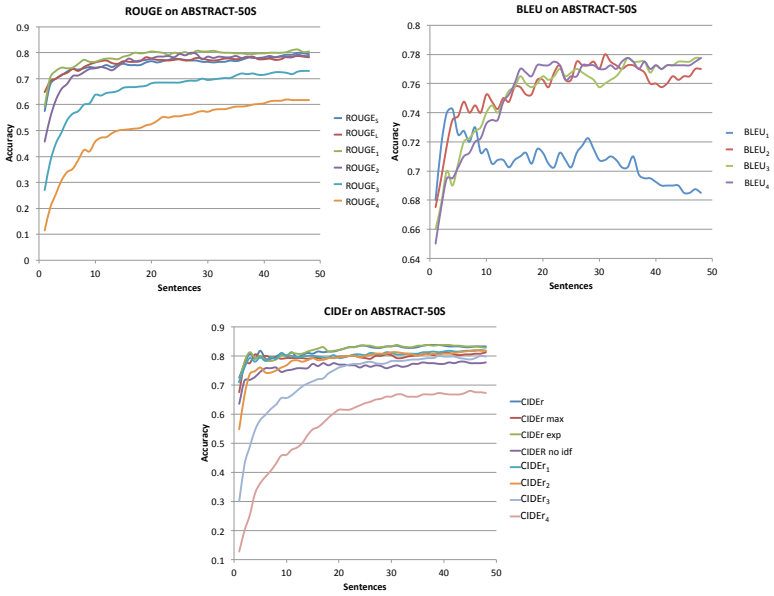


Figure A.7: Performance of different versions of metrics on ABSTRACT-50S

APPENDIX B

APPENDIX FOR CONTEXT-AWARE CAPTIONS FROM CONTEXT-AGNOSTIC SUPERVISION

We organize the appendix as follows:

- Sec. B.1: Analysis of performance as we consider unrelated images as distractors.
- Sec. B.2: Generating visual explanations (Hendricks et al. 2016b) adapted to the justification task.
- Sec. B.3: Architectural changes to the "Show, Attend, and Tell" image captioning model (Xu et al. 2015) for justification.
- Sec. B.4: Optimization details for justification speaker model.
- Sec. B.5: Choice of metrics for evaluating justification.
- Sec. B.6: CUB-Justify data collection details.
- Sec. B.7: Analysis of the RS(λ) baseline in more detail.

B.1 COCO Qualitative Results

COCO Qualitative Examples: Fig. B.1 shows more qualitative results on discriminative image captioning on the **hard confusion** split of the COCO dataset. Notice how our introspective speaker captions (denoted by IS), which model the context (distractor image) explicitly are often more discriminative, helping identify the target image more clearly than the baseline speaker approach (denoted by S). For example in the second row, our IS model generates the caption "a delta passenger jet flying through a clear blue sky", which is a more discriminative (and accurate) caption than the baseline caption "a large passenger jet flying through a blue sky", which applies to both the target and distractor images.

Effect of increasing distance: We illustrate how the quality of the discriminative captions from the introspective speaker (IS) approach varies as the distractor image becomes less

Figure B.1: Qualitative examples for discriminative image captioning (similar to Fig. 4.9). S (speaker) denotes examples from the standard image captioning model, which generates the same caption for the two images. Our method's outputs are shown as IS (introspective speaker). The target image is shown to the left and marked with a green border where our approach is accurate, as well as more discriminative. The second last example shows a case where our model is more discriminative, but inaccurate for the original target image and the last example shows a case where our caption is neither accurate not discriminative.

relevant to the target image (Fig. B.2). For the target image on the left, we show the 1-nearest neighbor (which has a very similar caption to the target image), the 10^{th}-nearest neighbor and a randomly selected distractor image. When we pick a random image to be the distractor, the generated discriminatve captions become less comprehensible, losing relevance as well as grammatical structure. This is consistent with our understanding of the introspective speaker (IS) formulation: modeling the context explicitly during inference helps discrimination when the context is relevant. When the context is not relevant, as with the randomly picked images, the original speaker model (S) is likely sufficient for discrimination.

Figure B.2: We show the target image (extreme left) and distractor images at varying distances (1 nearest neighbor, 10 nearest neighbor and random distractor), along with some generated captions. D denotes the distance between the target and distractor images in the FC7 space. The output of the speaker (S) is shown under the target image and the output of the introspective speaker considering each distractor image as context in turn, is shown under the corresponding distractor image. That is, the caption under each distractor image describes the target image distinguishing it from the distractor. Notice that our introspective speaker (IS) method often works well for 1 nearest neighbour and the 10^{th} nearest neighbor, but produces incomprehensible sentences when the distractor is irrelevant. Indeed, for a random distractor, we see that the baseline speaker outputs (S) are often sufficient for discrimination, which is intuitive.

B.2 Comparison to previous work on Generating Visual Explanations (Hendricks et al. 2016a)

Hendricks *et.al* (Hendricks et al. 2016a) propose a method to explain classification decisions to an end user by providing post-hoc rationalizations. Given a prediction from a classifier, this work generates a caption conditioned on the predicted class, and the original image. While Hendricks *et.al*aim to provide a rationale for a classification, we focus on a related but different problem of concept justification. Namely, we want to explain why an image contains a target class as opposed to a specific distractor class, while Hendircks *et.al*want to explain why a classifier thought an image contains a particular class. Thus, unlike the visual explanation task, it is intuitive that the justification task requires explicit reasoning about context. We verify this hypothesis, by first adapting the work of (Hendricks et al. 2016a) to our justification task, using it as a speaker, and then augmenting the speaker with our approach to construct an intropsective speakerm which accounts for context. Interestingly, we find that our introspective speaker approach helps improve the performance of generating visual explanations (Hendricks et al. 2016a) on justification.

The approach of Hendricks *et.al* (Hendricks et al. 2016a) differs from our setup in two important ways. Firstly, uses a stronger CNN, namely the fine-grained compact-bilinear pooling CNN which provides state-of-the-art performance on the CUB dataset. Secondly, to make the explanations more grounded in the class information, they also add a constraint to induce captions which are more specific to the class. This is achieved by using a policy gradient on a reward function that models $p(c|s)$ for a given sentence s and class c. Thus, in some sense the approach encourages the model to produce sentences that are highly discriminative of a given class against all other classes, as opposed to a particular distractor class that we are interested in for justification. Finally, the policy gradient is used in conjunction with standard maximum likelihood training to train the explanation model. At inference, the explanation model is run by conditioning the caption generation on the

predicted class.

We modify the inference setup of (Hendricks et al. 2016a) slightly to condition the caption generation on the *target* class for justification, as opposed to the predicted class for explanation. We call this the **vis-exp** approach. We then apply the emitter-suppressor beam search (at a beam size of 1, to be consistent with (Hendricks et al. 2016a)) to account for context, giving us an introspective visual explanation model (**vis-exp-IS**). Given the stronger image features and a more complicated training procedure involving policy gradients (hard to implement and tune in practice), the **vis-exp** approach achieves a strong CIDEr-D score of 20.36 with a standard error of 0.16 on our CUB-Justify test set. Note that this CUB-Justify test set is a strict subset of the test set from (Hendricks et al. 2016a). These results are better than those achieved with our semi-blind-IS(λ) CUB model, which is based on regular image features from VGG-16 implemented in the "Show, Attend and Tell" framework and uses standard log-likelihood training (Table. 4.2).

However, as mentioned before, the approach of (Hendricks et al. 2016a), similar to a baseline speaker S, cannot explicitly model context from a specific distractor class at inference. That is, while the approach reasons (through its training procedure) that given an image of a hummingbird, one should talk about its *long beak* (a discriminating feature for a hummingbird against all other birds), it cannot reason about a specific distractor class presented at inference. If the distractor class is another hummingbird with a long beak, we would want to avoid talking about the *long beak* in our justification. On the other hand, if the distractor class were a hummingbird with a shorter beak and there do exist such hummingbirds, then the *long beak* would be an important feature to mention in a justification. Clearly, this is non-trivial to realize without explicitly modeling context. Hence, intuitively, one would expect that incorporating context from the distractor class should help the justification task.

As explained previously, we implement our emitter-suppressor inference (Eqn. 4.8), on top of the **vis-exp** approach, yielding an **vis-exp-IS** approach. We sweep over the values

Table B.1: **CUB-Justify test results:** We compare **vis-exp** (Hendricks et al. 2016a) and our emitter-suppressor beam search implemented on top of **vis-exp**, namely **vis-exp-IS**. We see that we can achieve gains over the **vis-exp** approach by explicitly reasoning about context using our introspective speaker on the justification task. Error values are standard error of the mean.

Approach	CIDEr-D
vis-exp (Hendricks et al. 2016a)	20.36 ± 0.16
vis-exp-IS (ours)	21.52 ± 0.17

of λ on validation and find that the best performance is achieved at $\lambda = 0.9$. Plugging this value and evaluating on test, our **vis-exp-IS** approach achieves a CIDEr-D score of 21.52 with a standard error of 0.17 (Table. B.1). This is an improvement of 1.16 CIDEr-D. Our gains over **vis-exp** are lower than the gains on the IS(1) approach (reported in Table. 4.2), presumably because the **vis-exp** approach already captures a lot of the context-independent discriminative signals (*e.g.*, *long beak* for a hummingbird), due to policy gradient training. Overall though, these results provide further evidence that our emitter-suppressor inference scheme can be adapted to a variety of context-agnostic captioning models, to effectively induce context awareness during inference.

B.3 Architectures for Show, Attend, and Tell with Class Conditioning for CUB

We explain some minor modifications to the "Show, Attend and Tell" (Xu et al. 2015) image captioning model to condition it on the class label in addition to the image, for our experiments on CUB. Note that the explanation in this section is only for CUB – our COCO models are trained using the neuraltalk2 package[1] which implements the "Show and Tell" captioning model from Vinyals *et.al* (Vinyals et al. 2015). Our changes can be understood as three simple modifications aimed to use class information in the model. We first embed the class label (1 out of 200 classes for CUB) into a continuous vector $\mathbf{k} \in \mathbb{R}^D$, $D = 512$. The three changes then, on top of the Show, Attend, and Tell model (Xu et al. 2015) are as

[1] https://github.com/karpathy/neuraltalk2

follows:

- **Changes to initial LSTM state:** The original Show, Attend, and Tell model uses image annotation vectors a_i (i indexes spatial location), which are the outputs from a convolutional feature map to compute the initial cell and hidden states of the long-short term memory (LSTM) (c_0, h_0). The image annotation vector is averaged across spatial locations $\bar{\mathbf{a}} = \frac{1}{L}\sum_{i=1}^{L} \mathbf{a_i}$ and used to compute the initial state as follows:

$$\mathbf{c_0} = f_{init,c}(\bar{\mathbf{a}})$$

$$\mathbf{h_0} = f_{init,h}(\bar{\mathbf{a}})$$

  We modify this to also use the class embedding k to predict the initial state of the LSTM, by concatenating it with the averaged anntoation vector ($\bar{\mathbf{a}}$):

$$\mathbf{c_0} = f_{init,c}([\bar{\mathbf{a}}; \mathbf{k}])$$

$$\mathbf{h_0} = f_{init,h}([\bar{\mathbf{a}}; \mathbf{k}])$$

- **Changes to the LSTM recurrence:** "Show, Attend and Tell" computes a scalar attention α_{ti} at each location of the feature map and uses it to compute a context vector at every timestep $\hat{\mathbf{z_t}} = \phi(\{\alpha_{ti}, \mathbf{a_i}\})$ by attending on the image annotation a_i. It also embeds an input word y_t using an embedding matrix E and uses the previous hidden state h_t to compute the following LSTM recurrence at every timestep, producing outputs $\mathbf{i_t}$ (input gate), $\mathbf{f_t}$ (forget gate), $\mathbf{o_t}$ (output gate), $\mathbf{g_t}$ (input) (Eqn. 1, 2, 3

from (Xu et al. 2015)):

$$\begin{pmatrix} \mathbf{i_t} \\ \mathbf{f_t} \\ \mathbf{o_t} \\ \mathbf{g_t} \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} T \begin{pmatrix} E\mathbf{y_t} \\ \mathbf{h_{t-1}} \\ \hat{\mathbf{z}}_\mathbf{t} \end{pmatrix} \tag{B.1}$$

$$\mathbf{c_t} = \mathbf{f_t} \odot \mathbf{c_{t-1}} + \mathbf{i_t} \odot \mathbf{g_t} \tag{B.2}$$

$$\mathbf{h_t} = \mathbf{o_t} \odot \tanh(\mathbf{c_t}) \tag{B.3}$$

We use the class embeddings $\mathbf{k}$ in addition to the context vector $\hat{\mathbf{z}}_\mathbf{t}$ in Eqn. 1:

$$\begin{pmatrix} \mathbf{i_t} \\ \mathbf{f_t} \\ \mathbf{o_t} \\ \mathbf{g_t} \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} T' \begin{pmatrix} E\mathbf{y_t} \\ \mathbf{h_{t-1}} \\ \hat{\mathbf{z}}_\mathbf{t} \\ \mathbf{k} \end{pmatrix} \tag{B.4}$$

The remaining equations for the LSTM recurrence remain the same (Eqn. 2, 3).

- **Adding class information to the deep output layer:** "Show, Attend and Tell" uses a deep output layer to compute the output word distribution at every timestep, incorporating signals from the LSTM hidden state h_t, context vector $\hat{\mathbf{z}}_\mathbf{t}$ and the input word $\mathbf{y_t}$:

$$p(y_t) \propto \exp(L_o(E\mathbf{y_t} + L_h\mathbf{h_t} + L_z\mathbf{z_t}))$$

Here L_h, L_z are matrices used to project $\mathbf{h_t}$ and $\mathbf{z_t}$ to the dimensions of the word embeddings $E\mathbf{y_t}$ and L_o is the output layer which produces an output of the size of the vocabulary. Similar to the previous two adaptations, we use the class embedding $\mathbf{k}$ in

addition to the context vector $\hat{\mathbf{z}}_\mathbf{t}$ to predict the output at every timestep:

$$p(y_t) \propto \exp(L_o(E\mathbf{y_t} + L_h\mathbf{h_t} + L_z\mathbf{z_t} + L_k\mathbf{k}))$$

- **Blind models:** For implementing our class-only blind-IS(λ) model, we need to train a model that only uses the class to produce a sentence. For this, we drop the attention component from the model, which is equivalent to setting $\hat{z}_t$ and $\hat{\bar{a}}$ to zero for all our equations above and run the model using the class embedding $\mathbf{k}$.

B.4 Optimization Details

Our CUB captioning network is trained using *Rmsprop* with a batch size of 32 and a learning rate of 0.001. We decayed the learning rate on every 5 epochs of cycling through the training data. Our word embedding E embeds words into a 512 dimensional vector and we set LSTM hidden and cell state (h_0, c_0) sizes to 1800, similar to the "Show, Attend, and Tell" model on COCO. The rest of our design choices closely mirror the original work of (Xu et al. 2015), based on their implementation available at `https://github.com/kelvinxu/arctic-captions`. We will make our Tensorflow implementation of "Show, Attend, and Tell" publicly available.

B.5 Metrics for Justification

In this section, we expand more on our discussion on the choice of metrics for evaluating justification. In addition to the metrics we report in the paper, namely CIDEr-D (Vedantam, Lawrence Zitnick, and Parikh 2015) and METEOR (Banerjee and Lavie 2005), we also considered using the recently introduced SPICE (Anderson et al. 2016). The SPICE metric uses a dependency parser to extract a scene graph representation for the candidate and reference sentences and computes an F-measure between the scene graph representations. Given that the metric uses a dependency parser as an intermediate step, it is unclear how

Table B.2: **CUB-Justify validation results:** SPICE scores (higher the better) computed on validation set of CUB-Justify. Each model used its best λ value. Error values are standard error of the mean. IS(λ) outperforms the other methods by a good margin on SPICE.

Approach	SPICE
IS(λ)	**16.45 $\pm$ 0.12**
semi-blind-IS(λ)	15.59 $\pm$ 0.12
RS(λ)	14.69 $\pm$ 0.12
IS(1)	14.74 $\pm$ 0.12
blind-IS(λ)	15.7 $\pm$ 0.12

well it would scale to our justification task: some of the sentences from our model might be good justifications but may not be exactly grammatical. This is because our discriminative justifications emerge as a result of a tradeoff between high-likelihood sentences and discrimination (Eqn. 4.8). Note that this tradeoff is inherent since we don't have ground truth (well-formed) discriminative training data. Thus SPICE can be a problematic metric to use in our context. However, for the sake of completeness, we report SPICE numbers on validation, giving each approach access to its best λ value, in Table. B.2.

Although we outperform the baselines using the SPICE metric, in some corner cases we also found the SPICE metric scores to be slightly un-interpretable. For example, for the candidate sentence "this bird has a speckled belly and breast with a short pointy bill.", and reference sentences "This bird has a yellow eyebrow and grey auriculars", "This is a bird with yellow supercilium and white throat", the SPICE scores were higher than one would expect (0.30). For reference, an intuitively more related sentence "this is a grey and yellow bird with a yellow eyebrow." obtains a lower SPICE score of 0.28 for the same reference sentences. Further investigation revealed that the relation F-measure, which roughly measures if the two sentences encode the same relations, had a high score in these corner cases. We hypothesize that this inconcsistency in scores might be because SPICE uses soft similarity from WordNet for computing the F-measure, which might not be calibrated for this fine-grained domain, with specialized words such as *supercilium*, *auriculars etc*. As a result of these observations, we decided not to perform key evaluations with the SPICE

metric.

B.6 CUB-Justify Dataset Interface

We provide more details on the collection of the CUB-Justify dataset. We presented a target image from a selected target class to the workers along with a set of six distractor images, all belonging to one other distractor class. The distractor images were chosen at random from the validation, and test split of the CUB dataset we created for justification. Non-expert workers are unlikely to given have an explicit visual model of a given ditractor category, say Indigo Bunting. Thus the distractor images were shown to entail the concept of the distractor class for justification. The choice of the distractor classes is made based on the hierarchy we induce using the folk names of the birds. Given the target class, and the distractor class images, workers were asked to describe the target image in a manner that the sentence is not confusing with respect to the distractor images. Further, the workers were instructed that someone who reads the sentence should be able to recognize the target image, distinguishing it from the set of distractor images. In order to get workers to pay attention to all the images (and the intra-class invariances), they were not told explicitly that the distractor images all belonged to one other, unique, distractor class. For helping identify minute difference between images of birds, as well as enabling workers to write more accurate captions, we also showed them a diagram of the morphology of a bird (Fig. B.3). We also showed them a list of some other parts with examples not shown in the diagram, such as *eyeline*, *rump*, *eyering*, *etc*. The list of these words as well as examples, and the morphology diagram were picked based on consultation with an ornithology hobbyist. The workers were also explicitly instructed to describe only the target image, in an accurate manner, mentioning details that are present in the target image, as opposed to providing jusitifications that talk about features that are absent.

The initial rounds of data collection revealed some interesting corner cases that caused some ambiguity. For example, some workers were confused whether a part of the bird

should be called gray or white, because it could appear gray either because the part was white, and in shadow, or the part was actually gray. After these initial rounds of feedback, we proceeded to collect the entire dataset.

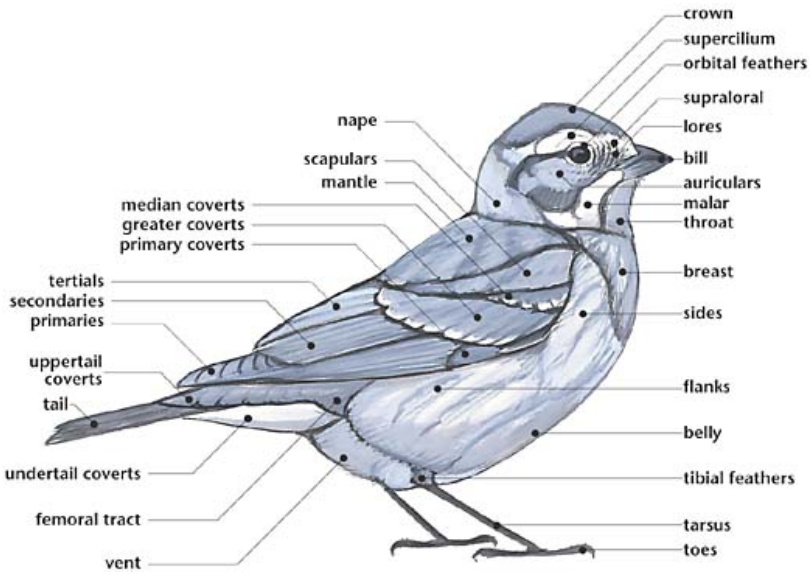


Figure B.3: A diagram of the morphology of a bird, labeling different parts. This diagram was shown to workers when getting justifications explaining why the image contains a target class, and not a distractor class.

B.7 Reasoning Speaker Performance Analysis

In this section, we provide more details on how the performance of our adaptation of Andreas, and Klein (Andreas and Klein 2016), namely the RS(λ) approach varies as we sweep over the number of samples we draw from the model for $\lambda = 0.3$, $\lambda = 0.5$, and $\lambda = 0.7$. We note that for $\lambda = 0.5$, the RS(λ) approach approaches the best performance from our IS(λ) approach as we draw 100 samples from the model (Fig. B.4). Interestingly, our IS(λ) model is only evaluated with a beam size of 10. Thus our model is able to perform more efficient search for discriminative sentences than a sampling, and re-ranking based approach like RS(λ). It is easy to note that, in case we were willing to spend time to enumerate over all exponentially-many sentences, we would find the optimal solution in worst case exponential time – most approximate inference techniques in such a setting offer

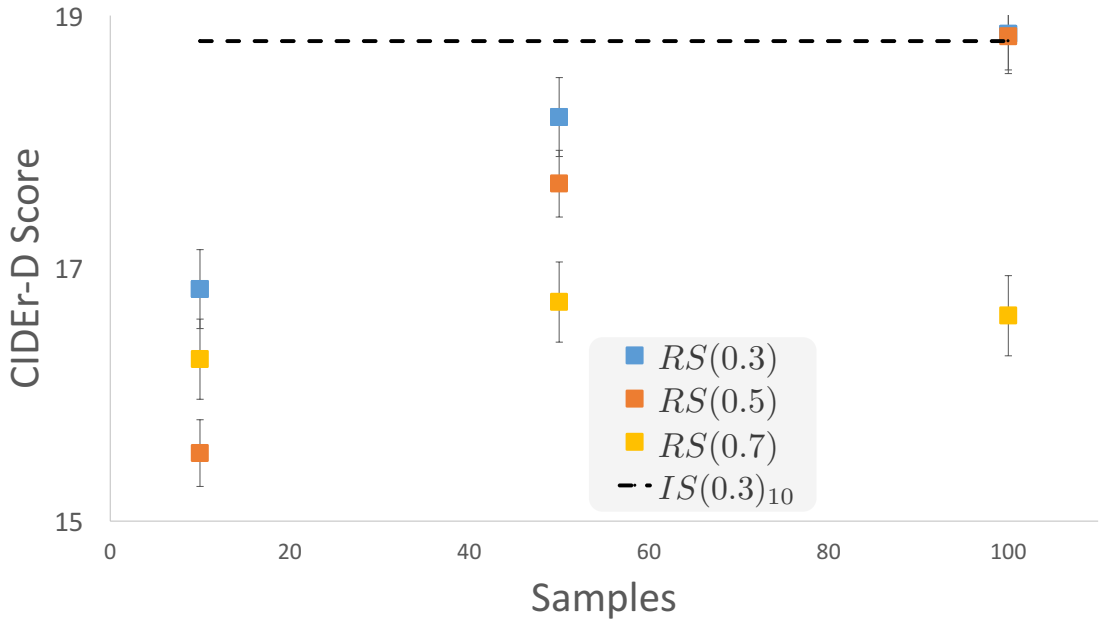


Figure B.4: We plot how the CIDEr-D score of the RS(λ) baseline (y-axis) varies with the number of samples (x-axis) for different values of λ. We see that for $\lambda = 0.5$, the performance of the RS(λ) method keeps increasing with the number of samples, reaching the performance of our IS(λ) approach at 100 samples. The IS(λ) method is shown for reference at a beam size of 10. Thus our approach (IS(λ)) is able to give better results for a lower computational cost.

a time vs. optimality tradeoff. Our approach seems to fit this tradeoff better than the RS(λ) approach based on this empirical evidence.

APPENDIX C

APPENDIX FOR LEARNING COMMONSENSE VIA VISUAL ABSTRACTION

C.1 Extracting Tuples from Sentences

As described in Chapter. 5, we build our VAL and TEST sets using the ReVerb information extraction system to extract our commonsense assertions. The ReVerb system segments the image into (typically) three chunks: primary object clause, relation clause and secondary object clause respectively. We do some post-processing to the ReVerb outputs to map them into our final t_P, t_R, and t_S tuples. We describe this post-processing below.

1. Get the Parts Of Speech (POS) tags for each input sentence.
2. Explore minor clauses in sentences by searching for one of the subordinating words ('because', 'although', 'unless', 'however', 'since') and extracting the shorter (minor) clause. In the minor clause, search for regular expression patterns: "*" is "*" to sample extra sentence chunks.
3. For all relation clauses, remove articles and pronoun instances.
4. For all relation clauses, remove the words "is" and "are".
5. For all primary and secondary clauses, remove pronouns, articles and adjectives.
6. Split to create new relations for each instance of "and". For example "Mike and Jenny *play* baseball" is converted to "Mike *play* baseball" and "Jenny *play* baseball"
7. Drop all relation clauses which contain a noun.
8. Perform lemmatization on all relation words. Lemmatization maps verbs to their root forms. Thus "plays" and "playing" are both mapped to "play".

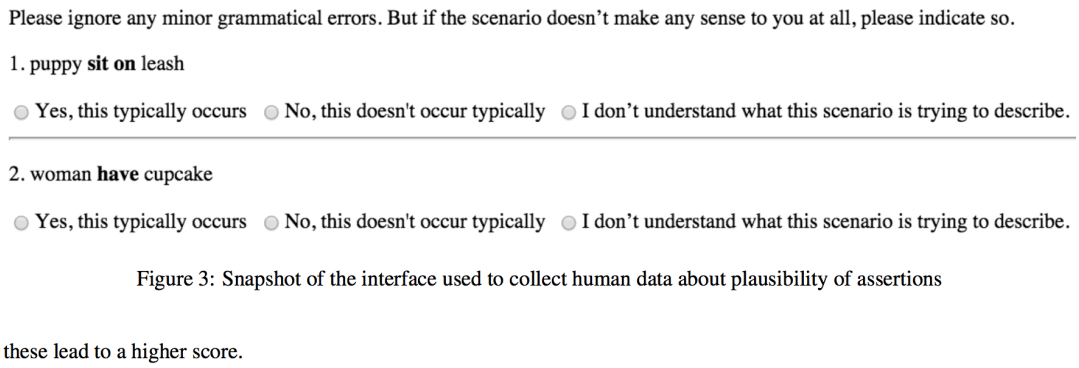


Figure 3: Snapshot of the interface used to collect human data about plausibility of assertions

these lead to a higher score.

Figure C.1: Snapshot of the interface used to collect human data about plausibility of assertions

9. Convert all plural nouns occuring in primary and secondary clauses to singular form. Also remove all instances of words ('group', 'couple', 'pair', 'bunch', 'crowd', 'team', 'two', 'three', 'four', 'five').

10. Remove all clauses with empty primary clause, secondary clause or relation clause to get the tuples.

C.2 Human Supervision for Feasibility of Assertions

We describe the interface (Figure C.1) we use for collecting ground truth plausibility of tuples or assertions. Workers on Amazon Mechanical Turk are shown a question and asked to rate if the scenario described by the assertion typically happens or not. We also give workers an option to tell us if the scenario described by the assertion makes no sense. We get 10 independent human responses for each such question, as described in Section 3 in the paper.

APPENDIX D

APPENDIX FOR VISUAL IMAGINATION

D.1 Appendix

D.1.1 Analysis of JMVAE objective

The JMVAE objective of (Suzuki, Nakayama, and Matsuo 2017b) has the form

$$J(\mathbf{x}, \mathbf{y}, \boldsymbol{\theta}, \boldsymbol{\phi}) = \mathrm{elbo}(\mathbf{x}, \mathbf{y}, \boldsymbol{\theta}, \boldsymbol{\phi}) - \alpha \left[\mathrm{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}, \mathbf{y}), q_{\boldsymbol{\phi}_y}(\mathbf{z}|\mathbf{y})) + \mathrm{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}, \mathbf{y}), q_{\boldsymbol{\phi}_x}(\mathbf{z}|\mathbf{x})) \right]$$

Let us focus on the $\mathrm{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}, \mathbf{y})|q_{\boldsymbol{\phi}_y}(\mathbf{z}|\mathbf{y}))$ term. Let $\mathcal{Y}$ be the set of unique labels (attribute vectors) in the training set, $\mathcal{X}_i$ be the indices of the images associated with label $\mathbf{y}_i$, and let $N_i = |\mathcal{X}_i|$ be the size of that set. Then we can write

$$\mathbb{E}_{\hat{p}(\mathbf{x},\mathbf{y})} \left[\mathrm{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}, \mathbf{y})|q_{\boldsymbol{\phi}_y}(\mathbf{z}|\mathbf{y})) \right] = \frac{1}{|\mathcal{Y}|} \sum_{i \in \mathcal{Y}} \frac{1}{N_i} \sum_{n \in \mathbf{X}_i} \mathrm{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_n, \mathbf{y}_i), q_{\boldsymbol{\phi}_y}(\mathbf{z}|\mathbf{y}_i)) \quad \text{(D.1)}$$

As explained in (Hoffman and Johnson 2016), we can rewrite this by treating the index $n \in \{1, \cdots, N_i\}$ as a random variable, with prior $q(n|\mathbf{y}_i) = 1/N_i$. Also, let us define the likelihood $q(\mathbf{z}|n, \mathbf{y}_i) = q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_n, \mathbf{y}_i)$. Using this notation, we can show that the above average KL becomes

$$\frac{1}{|\mathcal{Y}|} \sum_{i \in \mathcal{Y}} \left\{ \mathrm{KL}(q_{\boldsymbol{\phi}}^{\mathrm{avg}}(\mathbf{z}|\mathbf{y}_i)|q_{\boldsymbol{\phi}_y}(\mathbf{z}|\mathbf{y}_i)) + \log N_i - \mathbb{E}_{q_{\boldsymbol{\phi}_y}(\mathbf{z}|\mathbf{y}_i)} \left[\mathbb{H}(q(n|\mathbf{z}, \mathbf{y}_i)) \right] \right\} \quad \text{(D.2)}$$

where

$$q_{\boldsymbol{\phi}}^{\mathrm{avg}}(\mathbf{z}|\mathbf{y}_i) = \frac{1}{N_i} \sum_{n \in \mathcal{X}_i} q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_n, \mathbf{y}_i)$$

is the average of the posteriors for that concept, and $q(n|\mathbf{z}, \mathbf{y}_i)$ is the posterior over the indices for all the possible examples from the set $\mathcal{X}_i$, given that the latent code is $\mathbf{z}$ and the description is $\mathbf{y}_i$.

The $\mathrm{KL}(q_{\phi}^{\mathrm{avg}}(\mathbf{z}|\mathbf{y}_i)|q_{\phi_y}(\mathbf{z}|\mathbf{y}_i))$ term in eq. (D.2) tells us that JMVAE encourages the inference network for descriptions, $q_{\phi_y}(\mathbf{z}|\mathbf{y}_i)$, to be close to the average of the posteriors induced by each of the images $\mathbf{x}_n$ associated with $\mathbf{y}_i$. Since each $q_{\phi}(\mathbf{z}|\mathbf{x}_n, \mathbf{y}_i)$ is close to a delta function (since there is little posterior uncertainty when conditioning on an image), we are essentially requiring that $q_{\phi}(\mathbf{z}|\mathbf{y}_i)$ cover the embeddings of each of these images.

D.1.2 Details on the MNIST-A dataset

We created the MNIST-A dataset as follows. Given an image in the original MNIST dataset, we first sample a discrete scale label (big or small), an orientation label (clockwise, upright, and anti-clockwise), and a location label (top-left, top-right, bottom-left, bottom-right).

Next, we converted this vector of discrete attributes into a vector of continuous transformation parameters, using the procedure described below:

- **Scale:** For big, we sample scale values from a Gaussian centered at 0.9 with a standard deviation of 0.1, while for small we sample from a Gaussian centered at 0.6 with a standard deviation of 0.1. In all cases, we reject and draw a sample again if we get values outside the range $[0.4, 1.0]$, to avoid artifacts from upsampling or problems with illegible (small) digits.

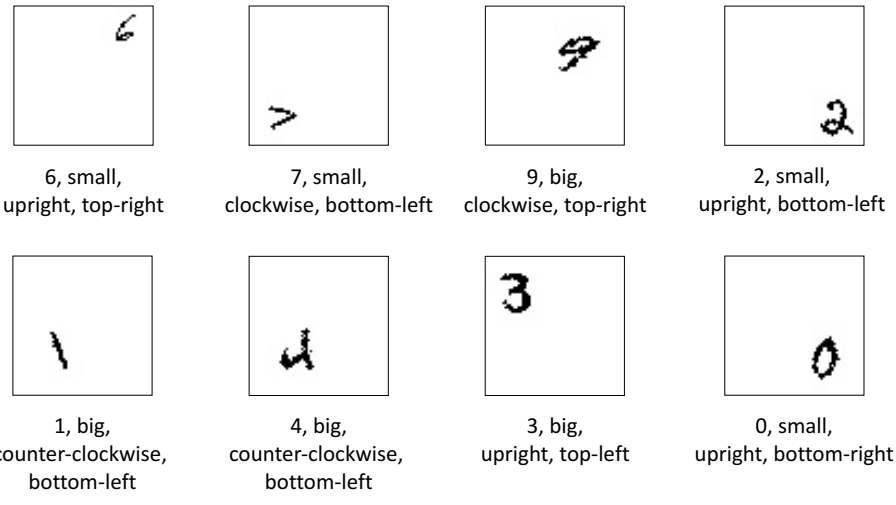
- **Orientation:** For the clockwise label, we sample the amount of rotation to apply for a digit from a Gaussian centered at +45 degrees, with a standard deviation of 10 degrees. For anti-clockwise, we use a Gaussian at -45 degrees, with a standard deviation of 10 degrees. For upright, we set the rotation to be 0 degrees always.

- **Location:** For location, we place Gaussians at the centers of the four quadrants in the image, and then apply an offset of `image_size/16` to shift the centers a bit towards

the corresponding corners. We then use a standard deviation of `image_size/16` and sample locations for centers of the digits. We reject and draw the sample again if we find that the location for the center would place the extremities of the digit outside of the canvas.

Finally, we generate the image as follows. We first take an empty black canvas of size `64x64`, rotate the original `28x28` MNIST image, and then scale and translate the image and paste it on the canvas. (We use bicubic interpolation for scaling and resizing the images.) Finally, we use the method of (Salakhutdinov and Murray 2008) to binarize the images. See Figure D.1 for example images generated in this way.

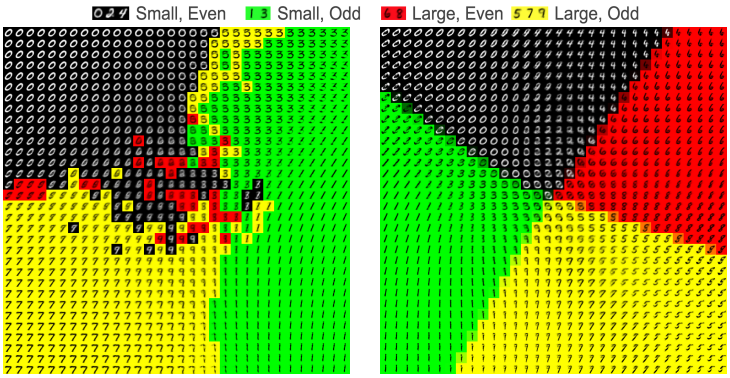
We repeat the above process of sampling labels, and applying corresponding transformations, to generate images 10 times for each image in the original MNIST dataset. Each trial samples labels from a uniform categorical distribution over the sample space for the corresponding attribute. Thus, we get a new MNIST-A dataset with 700,000 images from the original MNIST dataset of 70,000 images. We split the images into a train, val and test set of 85%, 5%, and 10% of the data respectively to create the IID split. To create the compositional split, we split the `10x2x3x4=240` possible label combinations by the sample train/val/test split, giving us splits of the dataset with non-overlapping label combinations.



Co

Figure D.1: Example binary images from our MNIST-A dataset.

D.1.3 β-VAE *vs.*Joint VAE



(a) (b)

Figure D.2: Visualization of the benefit of semantic annotations for learning a good latent space. Each small digit is a single sample generated from $p(x|z)$ from the corresponding point z in latent space. (a) β-VAE fit to images without annotations. The color of a point z is inferred from looking at the attributes of the training image that maps to this point of space using $q(z|x)$. Note that the red region (corresponding to the concept of large and even digits) is almost non existent. (b) Joint-VAE fit to images with annotations. The color of a point z is inferred from $p(y|z)$.

β-VAE (Higgins et al. 2017b) is an approach that aims to learn disentangled latent spaces. It does this by modifying the ELBO objective, so that it scales the $\mathrm{KL}(q(\mathbf{z}|\mathbf{x}), p(\mathbf{z}))$ term by a factor $\beta > 1$. This gives rise to disentangled spaces since the prior $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{I})$ is factorized (see (Achille and Soatto 2017) for details). However, to learn latent spaces that correspond to high level concepts, this is not sufficient: we need to use labeled data as well.

To illustrate this, we set up an experiment where we learn a 2d latent space for standard MNIST digit images, but where we replace the label with two binary attributes: parity (odd *vs.*even) and magnitude (value < 5 or $>= 5$). We call this dataset MNIST-2bit.

In Figure D.2(a), we show the results of fitting a 2d β-VAE model (Higgins et al. 2017b) to the images in MNIST-2bit, *ignoring the attributes*. We perform a hyperparameter sweep over β, and pick the one that gives the best looking latent space (this corresponds to a value of $\beta = 10$). At each point z in the latent 2d space, we show a single image sampled from $p(x|z)$. To derive the colors for each point in latent space, we proceed as follows: we embed each training image x (with label $y(x)$) into latent space, by computing $\hat{z}(x) = E_{q(z|x)}[z]$.

We then associate label $y(x)$ with this point in space. To derive the label for an arbitrary point z, we lookup the closest embedded training image (using ℓ_2 distance in z space), and use its corresponding label. We see that the latent space is useful for autoencoding (since the generated images look good), but it does not capture the relevant semantic properties of parity and magnitude. In fact, we argue that there is no way of forcing the model to learn a latent space that captures such high level conceptual properties from images alone.

In Figure D.2(b), we show the results of fitting a joint VAE model to MNIST-2bit, by optimizing $\mathrm{elbo}(x, y)$ on images and attributes (*i.e.*, we do not include the uni-modality $\mathrm{elbo}(x)$ and $\mathrm{elbo}(y)$ terms in this experiment.) Now the color codes are derived from $p(y|z)$ rather than using nearest neighbor retrieval. We see that the latent space autoencodes well, and also captures the 4 relevant types of concepts. In particular, the regions are all convex and linearly seperable, which facilitates the learning of a good imagination function $q(z|y)$, interpolation, retrieval, and other latent-space tasks.

A skeptic might complain that we have created an arbitrary partitioning of the data, that is unrelated to the appearance of the objects, and that learning such concepts is therefore "unnatural". But consider an agent interacting with an environment by touching digits on a screen. Suppose the amount of reward they get depends on whether the digit that they touch is small or big, or odd or even. In such an environment, it would be very useful for the agent to structure its internal representation to capture the concepts of magnitude and parity, rather than in terms of low level visual similarity. (In fact, (Scarf, Hayne, and Colombo 2011) showed that pigeons can learn simple numerical concepts, such as magnitude, by rewarding them for doing exactly this!) Language can be considered as the realization of such concepts, which enables agents to share useful information about their common environments more easily.

D.1.4 Details of the neural network architectures

As explained in the main paper, we fit the joint graphical model $p(x, y, z) = p(z)p(x|z)p(y|z)$ with inference networks $q(z|x, y)$, $q(z|x)$, and $q(z|y)$. Thus, our overall model is made up of three encoders (denoted with q) and two decoders (denoted with p). Across all models we use the exponential linear unit (ELU) which is a leaky non-linearity often used to train VAEs. We explain the architectures in more detail below.

MNIST-A model architecture

- Image decoder, $p(x|z)$: Our architecture for the image decoder exactly follows the standard DCGAN architecture from (Radford, Metz, and Chintala 2016), where the input to the model is the latent state of the VAE.

- Label decoder, $p(y|z)$: Our label decoder assumes a factorized output space $p(y|z) = \prod_{k \in \mathcal{A}} p(y_k|z)$, where y_k is each individual attribute. We parameterize each $p(y_k|z)$ with a two-layer MLP with 128 hidden units each. We apply a small amount of ℓ_2 regularization to the weight matrices.

- Image and Label encoder, $q(z|x, y)$: Our architecture (Figure D.3) for the image-label encoder first separately processes the images and the labels, and then concatenates them downstream in the network and then passes the concatenated features through a multi-layered perceptron. More specifically, we have convolutional layers which process image into $32, 64, 128, 16$ feature maps with strides $1, 2, 2, 2$ in the corresponding layers. We use batch normalization in the convolutional layers before applying the ELU non-linearity. On the label encoder side, we first encode the each attribute label into a 32d continuous vector and then pass each individual attribute vector through a 2-layered MLP with 512 hidden dimensions each. For example, for MNIST-A we have 4 attributes, which gives us 4 vectors of 512d. We then concatenate these vectors and pass it through a two layer MLP. Finally we concatenate this label feature with the image feature after the convolutional layers (after flattening the conv-features)

and then pass the result through a 2 layer MLP to predict the mean (μ) and standard deviation (σ) for the latent space gaussian. Following standard practice, we predict $\log \sigma$ for the standard deviation in order to get values which are positive.

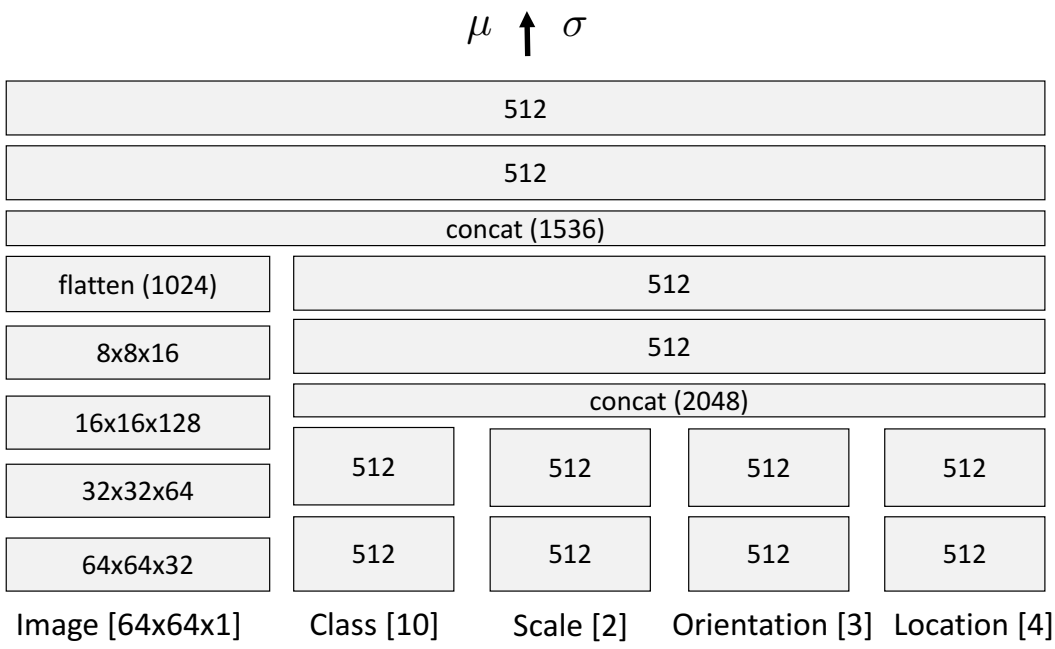
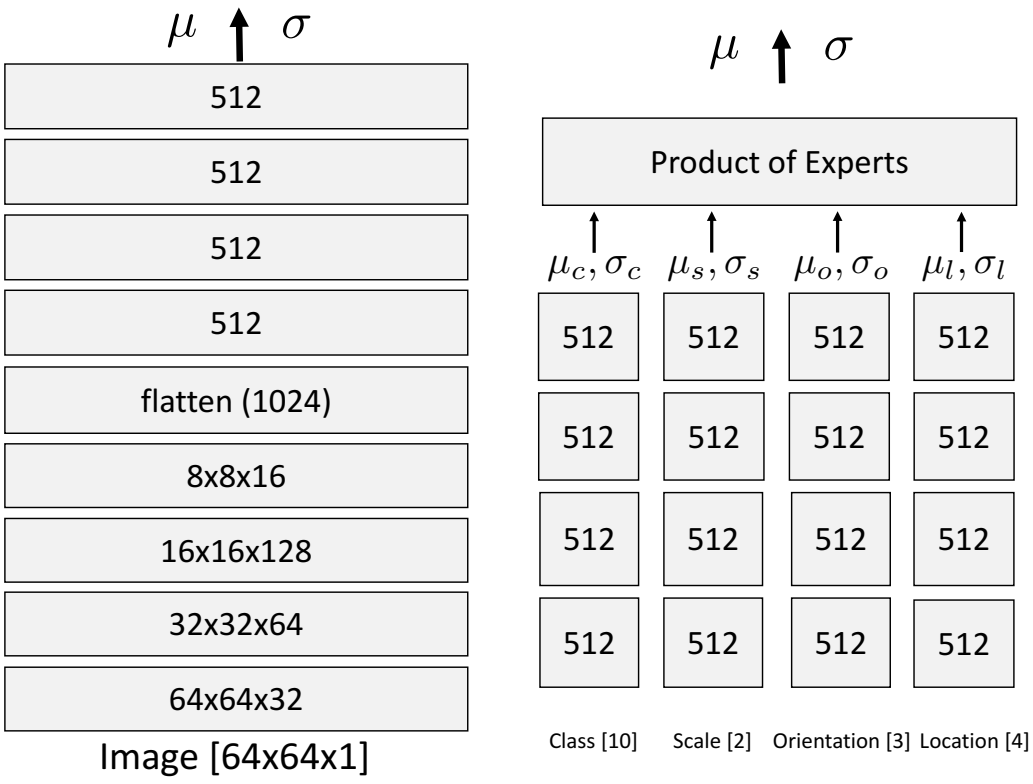


Figure D.3: Architecture for the $q(z|x, y)$ network in our JVAE models for MNIST-A. Images are `(64x64x1)`, class has 10 possible values, scale has 2 possible values, orientation has 3 possible values, and location has 4 possible values.

- Image encoder, $q(z|x)$: The image encoder (Figure D.4a) uses the same architecture to process the image as the image feature extractor in $q(z|x, y)$ network described above. After the conv-features, we pass the result through a 3-layer MLP to get the latent state mean and standard deviation vectors following the procedure described above.

- Label encoder, $q(z|y)$: The label encoder (Figure D.4b) part of the architecture uses the same design choices to process the labels as the label encoder part in the $q(z|x, y)$ network. After obtaining the initial, embeded attributes, we pass the result through four distinct 4-layered MLPs with 512 hidden dimensions each and then obtain the mean (μ) and $\log \sigma$ values for each attribute in the label set. The predicted μ and $\log \sigma$



(a) Architecture for the $q(z|x)$ network.

(b) Architecture for the $q(z|y)$ network.

Figure D.4: Archtectures for the single input inference networks for MNIST-A.

values for each attribute are fused together using the product of experts layer, which then outputs the final parameters (μ, $\log \sigma$) for the posterior.

MNIST-A Observation Classifier Model We next describe the architecuture of the observation classifier we use for evaluating the 3C's on the MNIST-A dataset. The observation classifier is a convolutional neural network, with the first convolutional layer with filters of size 5×5, and 32 channels, followed by a 2×2 pooling layer applied with a stride of 2. This is followed by another convolutional layer with 5×5 filter size and 64 output channels. This is followed by another 2×2 pooling layer of stride 2. After this, the network has four heads (corresponding to each attribute), each of which is an MLP with a single hidden layer (of size 1024), with dropout applied to the activations. The final layer of the MLP outputs the logits for classifying each attribute into the corresponding categorical labels associated with it. We train this model from scratch on the MNIST-A dataset using stochastic gradient descent, batch size of 64 and a learning rate of 10^{-4}.

CelebA model architecture Our design choices for CelebA closely mirror the models we built for MNIST-A. One primary difference is that we use a latent dimensionality of 18 in our CelebA experiments which matches the number of attributes we model. Meanwhile, the architectures of the image encoder, image decoder (*i.e.*DCGAN), are exactly identical to what is described above for MNIST-A execept that encoders take as input a 3-channel RGB image, while decoders produce a 3-channel output. We replace the Bernoulli likelihood with Quantized Normal likelihood (which is basically gaussian likelihood with uniform noise).

In terms of the label encoder $q(z|y)$, we follow Figure D.4b quite closely, except that we get as input 18 categorical (embedded) class labels as input, and we process the labels through a single hidden layer before concatenation and two hidden layers post concatenation (as opposed to two and four used in Figure D.4b).

Finally, the joint encoder $q(z|x, y)$, is again based heavily on Figure D.3 where we feed as input 18 labels as opposed to 4, process them through a single layer mlp of 512d, concatenate them, and then pass the result through a two hidden layer mlp of 512 d. At this

point we concatenate the result with the image feature through the image feature head in Figure D.3. Finally, we process the feature through another 512d single hidden layer mlp to produce the μ, σ values.

D.1.5 Outputs of observation classifier on generated images

fig. D.5 shows some images sampled from our TELBO model trained on MNIST-A. It also shows the attributes that are predicted by the attribute classifier. We see that the classifier often produces reasonable results that we as humans would also agree with. Thus, it acts as a reasonable proxy for humans classifying the labels for the generated images.

D.1.6 Hyperparamter Choices for TELBO, JMVAE, BiVCCA on MNIST-A

We discuss more hyperparameter choices for the different objectives and how they impact performance on the MNIST-A dataset. Across all the objectives we set λ_x=1, and vary λ_y. In addition, we also discuss how the private hyperparamter choices for each loss, γ for TELBO, α for JMVAE, as in (Wang, Lee, and Livescu 2016b)) and μ for BiVCCA affect performance. We use the JS-overall metric for picking hyperparameters, as explained in the main paper.

1. Effect of λ_y: We search for λ_y values in the set $\{1, 50, 100\}$ for all objectives. In general, we find the setting of λ_y in the elbo terms to be critical for good performance (especially on correctness). For example, at λ_y=1, we find that correctness numbers for the best performing TELBO model drop to 60.47 ($\pm$ 0.34) (from 82.08 ($\pm$ 0.56) at λ_y=50) on the validation set for `iid` queries. Similar trends can be observed for the JMVAE and BiVCCA objectives as well (with λ_y=10 being the best setting for BiVCCA, λ_y=50 for JMVAE). We have seen qualitative evidence which shows that the likelihood scaling for λ_y affects how disentangled the latent space is along the specified attributes. When the latent space is not grouped or organized as per high-level attributes (see fig. D.2 for example), the posterior distribution for a given concept

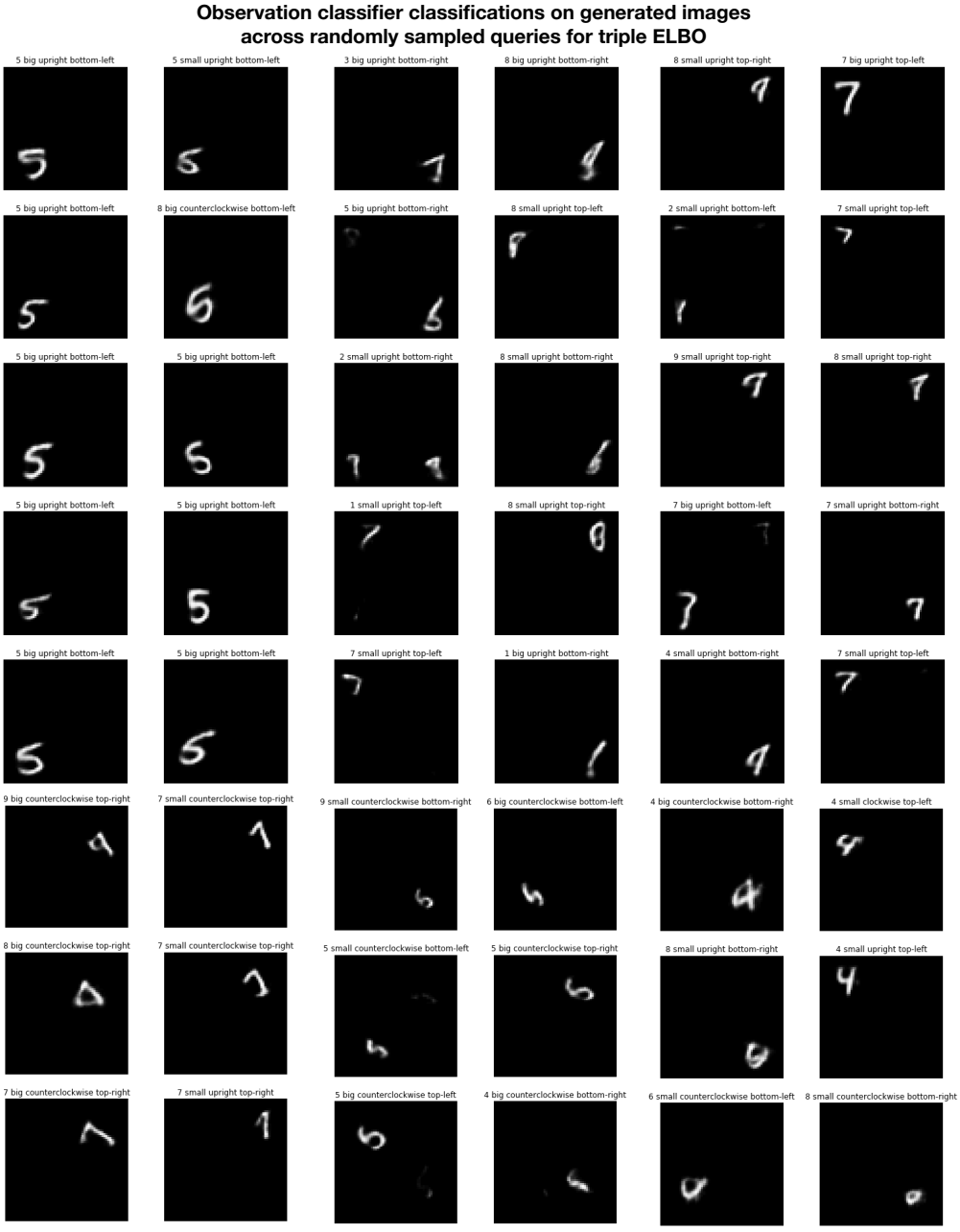


Figure D.5: Randomly sampled images from the TELBO model when fed randomly sampled concepts from the iid training set. We also show the outputs of the observation classifier for the images. Note that we visualize mean images above (since they tend to be more human interpretable) but the classifier is fed samples from the model. Figure best viewed by zooming in.

is multimodal, which is hard for a gaussian inference network $q(\mathbf{z}|\mathbf{y})$ to capture. This leads to poor correctness values.

2. Effect of γ: In addition to the λ_y scaling term which is common across all objectives, TELBO has a γ scaling factor which controls how we scale the $\log p(y|z)$ term in the $\mathrm{elbo}_{\gamma,1}(\mathbf{y}, \boldsymbol{\theta}_y, \boldsymbol{\phi}_y)$ term. We sweep values of $\{1, 50, 100\}$ for this parameter. In general, we find that the effect of this term is smaller on the performance than the λ_y term. Based on the setting of this parameter, we find that, for example, the correctness values for fully specified queries change from 82.08 ($\pm$0.56) at γ=50 to 80.27 ($\pm$0.38) at γ=1 on validation set for `iid` queries.

3. Effect of α: We generally find that α=1.0 works best for JMVAE across the different choices explored in (Wang, Lee, and Livescu 2016b), namely, $\{0.01, 0.1, 1.0\}$. For example, decreasing the value of α to 0.1 or 0.01 reduces correctness for fully sepcified queries from 85.63 ($\pm$0.29) to 77.58 ($\pm$0.23) at 0.1 and 74.57 ($\pm$0.44) at 0.01 respectively on the validation set for `iid` queries.

4. Effect of μ: For BiVCCA, we ran a search for μ over $\{0.3, 0.5, 0.7\}$, running each training experiment four times, and picked the best hyperparameter choice across the runs. We found that μ=0.7 was the best value, however the performance difference across different choices was not very large. Intuitively, higher values of μ should lead to improved performance compared to lower values of μ. This is because lower values of μ mean that we put more weight on the elbo term with a $q(\mathbf{z}|\mathbf{x})$ inference network than the one with a $q(\mathbf{z}|\mathbf{y})$ inference network, which results in sharper samples.

D.1.7 Compositional genralization on MNIST-A: Qualitative Results and Details

We next show some examples of compositional generalization on MNIST-A on a validation set of queries. For the compositinal experiments we reused the parameters of the best models on the iid splits for all the models, and trained the models for $\sim 160K$ iterations. All other

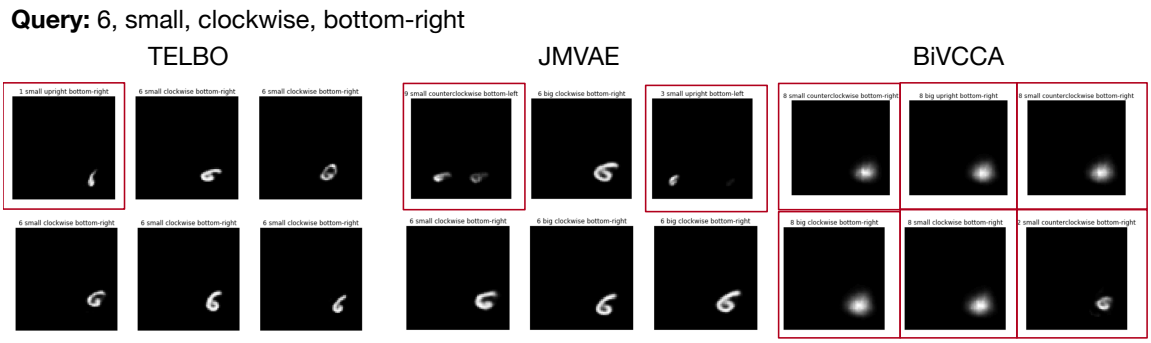


Figure D.6: Compositional generalization on MNIST-A. Models are given the unseen compositional query shown at the top and each of the three columns shows the mean of the image distribution generated by the models. Images marked with a red box are those that the observation classifier detected as being incorrect. We also show the classification result from the observation classifier on top of each image. We see that TELBO and JMVAE both do really well, while BiVCCA is substantially poorer.

design choices were the same. Figure D.6 shows some qualitative results.

D.1.8 Details on CelebA

CelebA consists of 202,599 face colored images and 40 attribute binary vectors. We use the version of this dataset that was used in (Perarnau et al. 2016); this uses a subset of 18 visually distinctive attributes, and preprocesses each image so they are aligned, cropped, and scaled down to 64 x 64. We use the official train and test partitions, 182K for training and 20K for testing. Note that this is an iid split, so the attribute vectors in the test set all occur in the training set, even though the images and people are unique. In total, the original dataset with 40 attributes specified a set of 96486 unique visual concepts, while our dataset of 18 attributes spans 3690 different visual concepts.

In section 6.4.2, we claim that our generations of "Bald" and "Female" images are from a compositionally novel concept. Our claim comes with a minor caveat/clarification: the concept `bald`=1 and `male`=0 does occur in 9 training examples, but they are all incorrect labelings, as shown in fig. D.7! Further, we see that the images generated from our model (shown in fig. 6.6) are qualitatively very different from any of the images here, showing that

Figure D.7: Set of all 9 images labelled as `bald=1` and `male=0` in the CelebA dataset. We can see that in all the cases the labels are inaccurate for the image, probably due to annotator error.

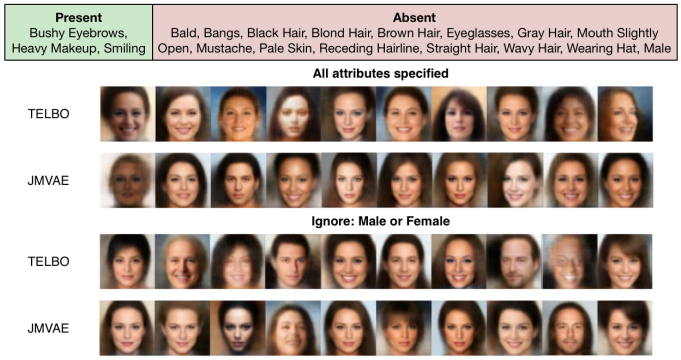


Figure D.8: **TELBO creates more diverse images than JMVAE.** At the top we show the set of attributes which are present and absent in the input query. Below, we show the results of generation with all the attributes specified, drawing 10 samples each. We see that both TELBO and JMVAE create accurate images satisfying the constraints. Note that the concept "male" is set to "absent" in the query, which in CelebA means that "female" is present. Next, we unspecify whether the image should contain a male or a female. We see that in this setting, TELBO has a better mixing of male and female images (fourth, sixth, eighth and ninth images in the third row are male), than JMVAE which just produces a single male image (the ninth image in the fourth row).

the model has not memorized these examples.

D.1.9 More results on CelebA

Finally, we show further qualitative examples of performance on the CelebA dataset. We focus on the TELBO and JMVAE objectives here, since BiVCCA generally produces poor samples (see fig. 6.6). fig. D.8 (middle) shows some example generations for the concept specified by the attributes (top). We see that both TELBO and JMVAE produce correct images when provided the full attribute queries (first two rows). However, when we stop specifying attribute "male" or "not male" (female), we see that TELBO provides more

diverse samples, spanning both male and female (compared to JMVAE). This ties into the explanation in section D.1.1, where we show how one can interpret JMVAE as optimizing for the $\mathrm{KL}(q_{\phi}^{\mathrm{avg}}(\mathbf{z}|\mathbf{y}_i)|q_{\phi_y}(\mathbf{z}|\mathbf{y}_i))$ to fit the unimodal inference network $q_{\phi_y}(\mathbf{z}|\mathbf{y}_i)$. Since JMVAE only reasons about the "aggregate" posterior as opposed to the prior (which TELBO reasons about), it has the tendency to generate less diverse samples when shown unseen concepts.

REFERENCES

Kitcher, Patricia (1988). “Marr’s Computational Theory of Vision”. In: *Philos. Sci.* 55.1, pp. 1–24.

Rosch, Eleanor (1999). “Principles of categorization”. In: *books.google.com*.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). “ImageNet Classification with Deep Convolutional Neural Networks”. In: *NIPS*. Ed. by F Pereira et al. Curran Associates, Inc., pp. 1097–1105.

Simonyan, Karen and Andrew Zisserman (2015). “Very Deep Convolutional Networks for Large-Scale Image Recognition”. In: *ICLR*.

Szegedy, Christian et al. (2015). “Going Deeper with Convolutions”. In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition*.

He, Kaiming et al. (2016). “Deep residual learning for image recognition”. In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 770–778.

Rumelhart, David E, Geoffrey E Hinton, and Ronald J Williams (Oct. 1986). “Learning representations by back-propagating errors”. In: *Nature* 323.6088, 323533a0.

Deng, Jia et al. (2009). “Imagenet: A large-scale hierarchical image database”. In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition*.

Donahue, Jeff et al. (Jan. 2014). “DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition”. In: *International Conference on Machine Learning*, pp. 647–655.

Sutskever, Ilya, Oriol Vinyals, and Quoc V Le (2014). “Sequence to Sequence Learning with Neural Networks”. In: *NIPS*. Ed. by Z Ghahramani et al. Curran Associates, Inc., pp. 3104–3112.

Karpathy, Andrej and Li Fei-Fei (2015). “Deep Visual-Semantic Alignments for Generating Image Descriptions”. In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition*.

Vinyals, Oriol et al. (2015). “Show and Tell: A Neural Image Caption Generator”. In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition*.

Chen, X and C L Zitnick (2015). “Mind’s eye: A recurrent visual representation for image caption generation”. In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition*.

Fang, Hao et al. (2015). “From Captions to Visual Concepts and Back”. In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition*.

Donahue, Jeff et al. (2015). *Long-term Recurrent Convolutional Networks for Visual Recognition and Description*.

Lin, T.Y. et al. (2014). “Microsoft COCO: Common Objects in Context”. In: *ECCV*.

Chen, Xinlei et al. (Apr. 2015). “Microsoft COCO Captions: Data Collection and Evaluation Server”. In: arXiv: 1504.00325 [cs.CV].

Kulkarni, Girish et al. (2011). “Baby talk: Understanding and generating image descriptions”. In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition*.

Liu, Siqi et al. (2017). “Improved Image Captioning via Policy Gradient optimization of SPIDEr”. In: *Intl. Conf. on Computer Vision*.

Andreas, Jacob and Dan Klein (2016). “Reasoning about pragmatics with neural listeners and speakers”. In: *Proc. Empirical Methods in Natural Language Processing*.

Harnad, Stevan (June 1990). “The symbol grounding problem”. In: *Physica D* 42.1, pp. 335–346.

Gordon, Jonathan and Benjamin Van Durme (Oct. 2013). “Reporting bias and knowledge acquisition”. In: *Proceedings of the 2013 workshop on Automated knowledge base construction*. ACM, pp. 25–30.

Mikolov, Tomas et al. (2013). “Distributed Representations of Words and Phrases and their Compositionality”. In: *NIPS*. Ed. by C J C Burges et al. Curran Associates, Inc., pp. 3111–3119.

Pennington, Jeffrey, Richard Socher, and Christopher Manning (2014). “Glove: Global vectors for word representation”. In: *Proc. Empirical Methods in Natural Language Processing*, pp. 1532–1543.

Xu, Ran et al. (2014a). “Improving Word Representations via Global Visual Context”. In: *NIPS Workshop on Learning Semantics*.

Lazaridou, Angeliki, Nghia The Pham, and Marco G Baroni (2015). “Combining Language and Vision with a Multimodal Skip-gram Model”. In: *HLT-NAACL*.

Vijayakumar, Ashwin, Ramakrishna Vedantam, and Devi Parikh (2017). “Sound-Word2Vec: Learning Word Representations Grounded in Sounds”. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, pp. 920–925.

Selvaraju, Ramprasaath R et al. (2017). “Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization”. In:

Fox, Chris and Shalom Lappin (2005). *Foundations of intensional semantics*. 1. publ. Malden, Mass. [u.a.]: Blackwell.

Dennett, Daniel C (Sept. 1983). “Intentional systems in cognitive ethology: The “Panglossian paradigm” defended”. In: *Behav. Brain Sci.* 6.3, pp. 343–355.

Searle, John R (Sept. 1980). “Minds, brains, and programs”. In: *Behav. Brain Sci.* 3.3, pp. 417–424.

Kingma, Diederik and Max Welling (2014a). “Auto-encoding variational Bayes”. In: *ICLR*.

LeCun, Y et al. (Nov. 1998). “Gradient-based learning applied to document recognition”. In: *Proc. IEEE* 86.11, pp. 2278–2324.

Bengio, Yoshua et al. (2003). “A Neural Probabilistic Language Model”. In: *J. of Machine Learning Research*.

Ren, Z et al. (2017). “Deep Reinforcement Learning-based Image Captioning with Embedding Reward”. In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition*.

Hochreiter, Sepp and Jurgen Schmidhuber (Nov. 1997). “Long Short-Term Memory”. In: *Neural Computation* 9.8, pp. 1735–1780.

Jordan, Michael I et al. (Nov. 1999). “An Introduction to Variational Methods for Graphical Models”. In: *Mach. Learn.* 37.2, pp. 183–233.

Farhadi, Ali et al. (2010). “Every Picture Tells a Story: Generating Sentences from Images”. In: *Proc. European Conf. on Computer Vision*.

Ordonez, Vicente, Girish Kulkarni, and Tamara L. Berg (2011). “Im2Text: Describing Images Using 1 Million Captioned Photographs”. In: *NIPS*.

Hodosh, Micah, Peter Young, and Julia Hockenmaier (2013). “Framing Image Description as a Ranking Task: Data, Models and Evaluation Metrics.” In: *J. Artif. Intell. Res. (JAIR)* 47, pp. 853–899.

Mitchell, Margaret, Xufeng Han, and Jeff Hayes (2012). “Midge: Generating Descriptions of Images”. In: *Proceedings of the Seventh International Natural Language Generation Conference*. INLG ’12. Utica, Illinois: Association for Computational Linguistics, pp. 131–133.

Yatskar, Mark et al. (2014). “See No Evil, Say No Evil: Description Generation from Densely Labeled Images”. In: *Proceedings of the Third Joint Conference on Lexical and Computational Semantics (*SEM 2014)*. Dublin, Ireland: Association for Computational Linguistics and Dublin City University, 110120.

Rohrbach, Marcus et al. (2013). “Translating Video Content to Natural Language Descriptions”. In: *Intl. Conf. on Computer Vision*.

Mao, Junhua et al. (2015a). “Explain Images with Multimodal Recurrent Neural Networks”. In: *ICLR*.

Xu, Kelvin et al. (2015). “Show, Attend and Tell: Neural Image Caption Generation with Visual Attention”. In: *ICML*.

Lu, Jiasen et al. (2017). “Knowing When to Look: Adaptive Attention via A Visual Sentinel for Image Captioning”. In:

Anderson, Peter et al. (July 2017). “Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering”. In: arXiv: 1707.07998 [cs.CV].

Everingham, M. et al. (Jan. 2015). “The Pascal Visual Object Classes Challenge: A Retrospective”. In: *International Journal of Computer Vision* 111.1, pp. 98–136.

Martin, D. et al. (2001). “A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics”. In: *Intl. Conf. on Computer Vision*. Vol. 2, pp. 416–423.

Scharstein, Daniel and Richard Szeliski (2002). “A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms”. In: *Int. J. Comput. Vision*.

Papineni, Kishore et al. (2002). “BLEU: A Method for Automatic Evaluation of Machine Translation”. In: *Proc. ACL*.

Lin, Chin-Yew (July 2004). “ROUGE: A Package for Automatic Evaluation of Summaries”. In: *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*. Ed. by Stan Szpakowicz Marie-Francine Moens. Barcelona, Spain: Association for Computational Linguistics, pp. 74–81.

Elliott, Desmond and Frank Keller (2014). “Comparing Automatic Evaluation Measures for Image Description”. In: *Proc. ACL*. Baltimore, Maryland: Association for Computational Linguistics, pp. 452–457.

Callison-burch, Chris and Miles Osborne (2006). “Re-evaluating the role of BLEU in machine translation research”. In: *In EACL*, pp. 249–256.

Anderson, Peter et al. (2016). “SPICE: Semantic Propositional Image Caption Evaluation”. In: *Proc. European Conf. on Computer Vision*.

Grice, H. (1975). “Logic and Conversation”. In: *Syntax and Semantics*. Academic Press.

Benotti, Luciana and David R. Traum (2009). “A computational account of comparative implicatures for a spoken dialogue agent”. In:

Vogel, Adam et al. (2013). “Emergence of Gricean Maxims from Multi-Agent Decision Theory”. In: *HLT-NAACL*.

Mordatch, Igor and Pieter Abbeel (Mar. 2017). “Emergence of Grounded Compositional Language in Multi-Agent Populations”. In: arXiv: `1703.04908 [cs.AI]`.

Das, Abhishek et al. (2017). “Learning Cooperative Visual Dialog Agents with Deep Reinforcement Learning”. In: *Intl. Conf. on Computer Vision*.

Lazaridou, Angeliki, Alexander Peysakhovich, and Marco Baroni (Dec. 2016). “Multi-Agent Cooperation and the Emergence of (Natural) Language”. In: *arXiv:1612.07182 [cs]*. arXiv: 1612.07182.

Wang, Sida I., Percy Liang, and Christopher D. Manning (2016). “Learning Language Games through Interaction”. In: *Proc. ACL*.

Tellex, Stefanie et al. (2014). “Asking for Help Using Inverse Semantics”. In: *Robotics: Science and Systems conference*.

Sadovnik, Amir et al. (2012). “Image description with a goal: Building efficient discriminating expressions for images”. In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition*.

Mao, Junhua et al. (2015b). “Generation and Comprehension of Unambiguous Object Descriptions”. In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition*.

Dong, Xin et al. (2014). “Knowledge vault: A web-scale approach to probabilistic knowledge fusion”. In: *KDD*. ACM, pp. 601–610.

Carlson, Andrew et al. (2010). “Toward an Architecture for Never-Ending Language Learning”. In: *AAAI*.

Etzioni, Oren et al. (2011). “Open Information Extraction: The Second Generation.” In: *IJCAI*. Vol. 11, pp. 3–10.

Bollacker, Kurt et al. (2008a). “Freebase: a collaboratively created graph database for structuring human knowledge”. In: *ACM SIGMOD*. ACM, pp. 1247–1250.

Sadeghi, Fereshteh, Santosh K Divvala, and Ali Farhadi (2015). "VisKE: Visual Knowledge Extraction and Question Answering by Visual Verification of Relation Phrases". In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1456–1464.

Divvala, S.K., A. Farhadi, and C. Guestrin (2014). "Learning Everything about Anything: Webly-Supervised Visual Concept Learning". In: *CVPR*.

Li, Yikang et al. (2017). "Scene Graph Generation from Objects, Phrases and Region Captions". In: *Intl. Conf. on Computer Vision*.

Lu, Cewu et al. (2016). "Visual Relationship Detection with Language Priors". In: *Proc. European Conf. on Computer Vision*.

Divvala, Santosh Kumar et al. (2009). "An empirical study of context in object detection". In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition*.

Fouhey, David F et al. (2012). "People watching: Human actions as a cue for single view geometry". In: *Proc. European Conf. on Computer Vision*.

Berg, Alexander C et al. (2012). "Understanding and predicting importance in images". In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition*.

Hays, James and Alexei A. Efros (2008). "IM2GPS: estimating geographic information from a single image". In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition*.

Khosla, Aditya et al. (2014). "Looking Beyond the Visible Scene". In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition*.

Pickup, L.C. et al. (2014). "Seeing the Arrow of Time". In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition*.

Pirsiavash, Hamed, Carl Vondrick, and Antonio Torralba (2014). "Inferring the Why in Images". In: *CoRR* abs/1406.5472.

Zhu, Yuke, Alireza Fathi, and Li Fei-Fei (2014). "Reasoning about Object Affordances in a Knowledge Base Representation". In: *Proc. European Conf. on Computer Vision*.

Johnson, Justin et al. (2015). "Image Retrieval using Scene Graphs". In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition*.

Krishna, Ranjay et al. (2016). "Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations". In:

Chen, Xinlei, Abhinav Shrivastava, and Abhinav Gupta (2013). "NEIL: Extracting visual knowledge from web data". In: *Intl. Conf. on Computer Vision*.

Chao, Yu-Wei et al. (2015). “Mining Semantic Affordances of Visual Object Categories”. In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition*.

Lin, Xiao and Devi Parikh (2015). “Don’t Just Listen, Use Your Imagination: Leveraging Visual Common Sense for Non-Visual Tasks”. In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition*.

Zitnick, C. Lawrence and Devi Parikh (2013). “Bringing Semantics Into Focus Using Visual Abstraction”. In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition*.

Zitnick, C.L., R. Vedantam, and D. Parikh (2016). “Adopting Abstract Images for Semantic Scene Understanding”. In: *IEEE PAMI*.

Zitnick, C Lawrence, Devi Parikh, and Lucy Vanderwende (2013). “Learning the visual interpretation of sentences”. In: *Intl. Conf. on Computer Vision*.

Fouhey, David F and C Lawrence Zitnick (2014). “Predicting Object Dynamics in Scenes”. In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition*.

Antol, Stanislaw, C Lawrence Zitnick, and Devi Parikh (2014). “Zero-Shot Learning via Visual Abstraction”. In: *Proc. European Conf. on Computer Vision*.

Wu, Jiajun, Joshua B Tenenbaum, and Pushmeet Kohli (2017). “Neural Scene De-rendering”. In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition*.

Zhang, Peng et al. (2015). “Yin and Yang: Balancing and Answering Binary Visual Questions”. In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition*.

Antol, Stanislaw et al. (2015). “VQA: Visual Question Answering”. In: *Intl. Conf. on Computer Vision*.

Ortiz, Luis Gilberto Mateos, Clemens Wolff, and Mirella Lapata (2015). “Learning to Interpret and Describe Abstract Scenes”. In: *NAACL*, pp. 1505–1515.

Kottur, Satwik et al. (2015). “Visual Word2Vec (vis-w2v): Learning Visually Grounded Word Embeddings Using Abstract Scenes”. In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition*.

Rocktäschel, Tim et al. (2016). “Reasoning about Entailment with Neural Attention”. In:

Lample, Guillaume et al. (2016). “Neural Architectures for Named Entity Recognition”. In:

Gao, Haoyuan et al. (2015). “Are You Talking to a Machine? Dataset and Methods for Multilingual Image Question”. In: *NIPS*. Ed. by C. Cortes et al.

Irsoy, Ozan and Claire Cardie (2014). “Deep Recursive Neural Networks for Compositionality in Language”. In: *NIPS*. Ed. by Z. Ghahramani et al.

Xu, Ran et al. (2014b). “Improving Word Representations via Global Visual Context”. In:

Silberer, Carina, Vittorio Ferrari, and Mirella Lapata (2013). “Models of Semantic Representation with Visual Attributes”. In: *Proc. ACL*.

Lopopolo, Alessandro and Emiel van Miltenburg (2015). “Sound-based distributional models”. In: *IWCS 2015*.

Kiela, Douwe and Stephen Clark (2015). “Multi-and Cross-Modal Semantics Beyond Vision: Grounding in Auditory Perception.” In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Bruni, Elia, Nam-Khanh Tran, and Marco Baroni (2014). “Multimodal Distributional Semantics.” In: *Journal of Artificial Intelligence Research (JAIR)*.

Melamud, Oren et al. (2016). “The role of context types and dimensionality in learning word embeddings”. In: *arXiv preprint arXiv:1601.00893*.

Goodfellow, Ian J et al. (2014a). “Generative Adversarial Networks”. In: *NIPS*.

Higgins, Irina et al. (2017a). “beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework”. In: *ICLR*.

Chen, Xi et al. (2016). “InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets”. In: *NIPS*.

Oord, Aaron van den et al. (2016). “Conditional Image Generation with PixelCNN Decoders”. In:

Kingma, Diederik P et al. (2014). “Semi-Supervised Learning with Deep Generative Models”. In:

Yan, Xinchen et al. (2016a). “Attribute2Image: Conditional Image Generation from Visual Attributes”. In:

Reed, Scott et al. (2016a). “Generative Adversarial Text to Image Synthesis”. In: *ICML*.

Isola, Phillip et al. (2017a). “Image-to-Image Translation with Conditional Adversarial Networks”. In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition*.

Wang, Weiran, Honglak Lee, and Karen Livescu (2016a). “Deep Variational Canonical Correlation Analysis”. In: *arXiv*.

Suzuki, Masahiro, Kotaro Nakayama, and Yutaka Matsuo (2017a). “Joint Multimodal Learning with Deep Generative Models”. In:

Tamuz, Omer et al. (2011). “Adaptively Learning the Crowd Kernel”. In: *ICML*. USA.

Bogacz, Rafal et al. (2006). “The physics of optimal decision making: a formal analysis of models of performance in two-alternative forced-choice tasks.” In: *Psychological review* 113 4, pp. 700–65.

Lavie, Alon and Michael J. Denkowski (2009). “The Meteor metric for automatic evaluation of machine translation”. In: *Machine Translation* 23, pp. 105–115.

Rashtchian, Cyrus et al. (2010). “Collecting Image Annotations Using Amazon’s Mechanical Turk”. In: *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon’s Mechanical Turk*. CSLDAMT ’10. Stroudsburg, PA, USA: Association for Computational Linguistics.

Robertson, Stephen E. (2004). “Understanding inverse document frequency: on theoretical arguments for IDF”. In: *Journal of Documentation* 60, pp. 503–520.

Plummer, Bryan A. et al. (2015). “Flickr30k Entities: Collecting Region-to-Phrase Correspondences for Richer Image-to-Sentence Models”. In: *Intl. Conf. on Computer Vision*.

Denkowski, Michael J. and Alon Lavie (2014). “Meteor Universal: Language Specific Translation Evaluation for Any Target Language”. In: *WMT@ACL*.

Wah, C. et al. (2011). *The Caltech-UCSD Birds-200-2011 Dataset*. Tech. rep. CNS-TR-2011-001. California Institute of Technology.

Reed, Scott E. et al. (2016c). “Learning Deep Representations of Fine-Grained Visual Descriptions”. In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 49–58.

Lee, Kai-Fu, Hsiao-Wuen Hon, and Raj Reddy (1990). “An overview of the SPHINX speech recognition system”. In: *Intl. Conf. on Acoustics, Speech and Signal Proc.* Vol. 38, pp. 35–45.

Vijayakumar, Ashwin K. et al. (2018). “Diverse Beam Search: Decoding Diverse Solutions from Neural Sequence Models”. In: *AAAI*.

Hendricks, Lisa Anne et al. (2016a). “Generating Visual Explanations”. In: *Proc. European Conf. on Computer Vision*.

Vedantam, Ramakrishna, C. Lawrence Zitnick, and Devi Parikh (2015). "CIDEr: Consensus-Based Image Description Evaluation". In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition*.

Karpathy, Andrej. *Neuraltalk2 Image Captioning*. `https://github.com/karpathy/neuraltalk2`.

Davis, Randall, Howard E. Shrobe, and Peter Szolovits (1993). "What Is a Knowledge Representation?" In: *AI Magazine* 14, pp. 17–33.

Hoffart, Johannes et al. (2013). "YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia". In: *Artif. Intell.* 194, pp. 28–61.

Lehmann, Jens et al. "DBpedia - A large-scale, multilingual knowledge base extracted from Wikipedia". In:

Bollacker, Kurt D. et al. (2008b). "Freebase: a collaboratively created graph database for structuring human knowledge". In: *SIGMOD Conference*.

Miller, George A. (1992). "WORDNET: A Lexical Database for English". In: *Commun. ACM* 38, pp. 39–41.

Singh, Push et al. (2002). "Open Mind Common Sense: Knowledge Acquisition from the General Public". In: *CoopIS/DOA/ODBASE*.

Speer, Robert and Catherine Havasi (2012). "Representing General Relational Knowledge in ConceptNet 5". In: *LREC*.

Jas, M and D Parikh (June 2015). "Image specificity". In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 2727–2736.

Loper, Edward and Steven B Bird (2002). "NLTK: The Natural Language Toolkit". In: *CoRR* cs.CL/0205028.

Morzycki, Marcin (2015). *Modification*. Key Topics in Semantics and Pragmatics. Cambridge University Press.

Hinton, Geoffrey E (Aug. 2002a). "Training products of experts by minimizing contrastive divergence". In: *Neural Comput.* 14.8, pp. 1771–1800.

Young, Peter et al. (2014a). "From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions". In: *Trans. Assoc. Comp. Ling.* Pp. 67–78.

Tenenbaum, J. B. (1999b). "Bayesian modeling of human concept learning". In: *NIPS*.

Liu, Ziwei et al. (2015). “Deep Learning Face Attributes in the Wild”. In: *Intl. Conf. on Computer Vision*.

Wang, Weiran, Honglak Lee, and Karen Livescu (Nov. 2016b). “Deep Variational Canonical Correlation Analysis”. In: *arXiv [cs.LG]*.

Kingma, Diederik P and Max Welling (2014b). “Auto-encoding variational Bayes”. In: *ICLR*.

Achille, Alessandro and Stefano Soatto (2017). “On the Emergence of Invariance and Disentangling in Deep Representations”. In: *arXiv*. arXiv: 1706.01350 [cs.LG].

Higgins, Irina et al. (2017b). “beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework”. In: *ICLR*.

Radford, Alec, Luke Metz, and Soumith Chintala (2016). “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks”. In: *ICLR*.

Pu, Yunchen et al. (2016). “Variational Autoencoder for Deep Learning of Images, Labels and Captions”. In: *NIPS*, pp. 2352–2360.

Williams, Christopher K I and Charlie Nash (Jan. 2018). “Autoencoders and Probabilistic Inference with Missing Data: An Exact Solution for The Factor Analysis Case”. In: arXiv: 1801.03851 [cs.LG].

Bouchacourt, Diane, Ryota Tomioka, and Sebastian Nowozin (2018). “Multi-Level Variational Autoencoder: Learning Disentangled Representations from Grouped Observations”. In: *AAAI*.

Hinton, Geoffrey E (Aug. 2002b). “Training products of experts by minimizing contrastive divergence”. In: *Neural Comput.* 14.8, pp. 1771–1800.

Williams, C K I and F V Agakov (May 2002). “Products of gaussians and probabilistic minor component analysis”. In: *Neural Computation* 14.5, pp. 1169–1182.

Goodfellow, Ian J et al. (2014b). “Generative Adversarial Networks”. In: *NIPS*.

Theis, Lucas, Aäron van den Oord, and Matthias Bethge (2016). “A note on the evaluation of generative models”. In: *ICLR*.

Salimans, Tim et al. (2016). “Improved Techniques for Training GANs”. In: *arXiv*. arXiv: 1606.03498 [cs.LG].

Yan, Xinchen et al. (2016b). “Attribute2Image: Conditional Image Generation from Visual Attributes”. In: *ECCV*.

Reed, Scott et al. (2016b). “Generative Adversarial Text-to-Image Synthesis”. In: *ICML*.

Isola, Phillip et al. (2017b). “Image-to-Image Translation with Conditional Adversarial Networks”. In: *CVPR*.

Wang, Weiran, Honglak Lee, and Karen Livescu (2016c). “Deep Variational Canonical Correlation Analysis”. In: *arXiv*.

Suzuki, Masahiro, Kotaro Nakayama, and Yutaka Matsuo (2017b). “Joint Multimodal Learning with Deep Generative Models”. In: *ICLR Workshop*.

Hoffman, M D and Johnson (2016). “Elbo surgery: yet another way to carve up the variational evidence lower bound”. In: *NIPS Workshop on Advances in Approximate Bayesian Inference*.

Higgins, Irina et al. (2017c). “SCAN: Learning Abstract Hierarchical Compositional Visual Concepts”. In: *Arxiv*. arXiv: 1707.03389 [stat.ML].

Hoffman, Matthew D (2017). “Learning Deep Latent Gaussian Models with Markov Chain Monte Carlo”. In: *ICML*, pp. 1510–1519.

Vilnis, Luke and Andrew McCallum (2015). “Word Representations via Gaussian Embedding”. In: *ICLR*.

Athiwaratkun, Ben and Andrew Gordon Wilson (2017). “Multimodal Word Distributions”. In: *Proc. ACL*.

Mukherjee, Tanmoy and Timothy Hospedales (2016). “Gaussian visual-linguistic embedding for zero-shot recognition”. In: *Proc. Empirical Methods in Natural Language Processing*.

Ren, Zhou et al. (2016). “Joint Image-Text Representation by Gaussian Visual-Semantic Embedding”. In: *Proc. ACM conf. on Multimedia*.

Young, Peter et al. (2014b). “From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions”. In: *Trans. Assoc. Comp. Ling.* Pp. 67–78.

Vendrov, Ivan et al. (2016). “Order-Embeddings of Images and Language”. In: *ICLR*.

Atzmon, Yuval et al. (2016). “Learning to generalize to new compositions in image understanding”. In: *ArXiv*. arXiv: 1608.07639 [cs.CV].

Johnson, Justin et al. (2017). “CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning”. In: *CVPR*.

Agrawal, A. et al. (2017). “C-VQA: A Compositional Split of the Visual Question Answering (VQA) v1.0 Dataset”. In: *ArXiv*. arXiv: 1704.08243 [cs.CV].

Krause, A., A. Singh, and C. Guestrin (2008). “Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies”. In: *J. of Machine Learning Research* 9, 235284.

Nemhauser, G., L. Wolsey, and M. Fisher (1978). “An analysis of approximations for maximizing submodular set functions”. In: *Mathematical Programming* 14, 265294.

Kingma, Diederik and Jimmy Ba (2015). “Adam: A Method for Stochastic Optimization”. In: *ICLR*.

Wang, Zhou et al. (Apr. 2004). “Image quality assessment: from error visibility to structural similarity”. In: *IEEE Trans. Image Process.* 13.4, pp. 600–612.

Perarnau, Guim et al. (2016). “Invertible Conditional GANs for image editing”. In: *NIPS Workshop on Adversarial Training*.

Tenenbaum, J. (1999a). “A Bayesian framework for concept learning”. PhD thesis. MIT.

Jia, Yangqing et al. (2013). “Visual Concept Learning: Combining Machine Vision and Bayesian Generalization on Concept Hierarchies”. In: *NIPS*.

Hendricks, Lisa Anne et al. (2016b). “Generating Visual Explanations”. In: *ECCV*.

Banerjee, Satanjeev and Alon Lavie (2005). “METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments”. In: pp. 65–72.

Salakhutdinov, Ruslan and Iain Murray (2008). “On the Quantitative Analysis of Deep Belief Networks”. In: *ICML*.

Scarf, Damian, Harlene Hayne, and Michael Colombo (2011). “Pigeons on par with primates in numerical competence”. In: *Science* 334.6063, p. 1664.

LIST OF PUBLICATIONS

1. **Generative Models of Visually Grounded Imagination**: Ramakrishna Vedantam, Ian Fischer, Jonathan Huang, Kevin Murphy. *International Conference on Learning Representations (ICLR), 2018*

2. **Grad-CAM: Why did you say that? Visual Explanations from Deep Networks via Gradient-based Localization**: Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, Dhruv Batra. *International Conference on Computer Vision (ICCV), 2017*

   Also presented at *NIPS Workshop on Interpretable Machine Learning in Complex Systems, 2016*

3. **Sound-Word2Vec: Learning Word Representations Grounded in Sounds**: Ashwin K. Vijayakumar, Ramakrishna Vedantam, Devi Parikh. *Conference on Empirical Methods in Natural Language Processing (EMNLP), 2017*

4. **Counting Everyday Objects in Everyday Scenes**: Prithvijit Chattopadhyay*, Ramakrishna Vedantam*, Ramprasaath R. Selvaraju, Dhruv Batra, Devi Parikh. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017* ***(Spotlight)***

5. **Context-aware Captions from Context-agnostic Supervision**: Ramakrishna Vedantam, Samy Bengio, Kevin Murphy, Devi Parikh, Gal Chechik. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017* ***(Spotlight)***

   Also presented as an Oral at the *Bay Area Machine Learning Symposium (BayLearn), 2017*.

6. **Adopting Abstract Images for Semantic Scene Understanding**: C. Lawrence Zitnick, Ramakrishna Vedantam, and Devi Parikh. *Special Issue on the best papers at*

the 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)
IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 2016

7. **Visual Word2Vec (`vis-w2v`): Learning Visually Grounded Word Embeddings using Abstract Scenes**: Satwik Kottur*, Ramakrishna Vedantam*, José Moura, and Devi Parikh. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016*

8. **Learning Common Sense through Visual Abstraction**: Ramakrishna Vedantam*, Xiao Lin*, Tanmay Batra, C. Lawrence Zitnick, and Devi Parikh. *IEEE International Conference on Computer Vision (ICCV), 2015*
   Also presented as an oral at *1st Workshop on Object Understanding for Interaction*, colocated with *ICCV, 2015*

9. **CIDEr: Consensus-based Image Description Evaluation**: Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015*

* Equal Contribution